

Apple



Assembly Line

\$1.80

Volume 7 -- Issue 7

April, 1987

In This Issue...

80-Column Screen Display Subroutine.	2
Revision to DOS 3.3 for UniDisk 3.5.	13
A PER-fect Tool for 65816 Position-Independent Code.	14
Simple Line-Input Subroutine	19
Friendly Decimal Print Subroutine.	22
Note from American Foundation for the Blind.	26
Some Handy MOVE Macros	28
New Supplement to "Beneath Apple ProDOS"	30
A Note about Chessmaster 2000.	32

ByteStalker Software Analyzer Card

Old-fashioned computers always came with a front panel, loaded with lights and switches. You could usually stop a program, single-step it, and even set a breakpoint address to halt execution at a particular address. Personal computers don't include front panels, but now you can add part of one to your Apple. J. Stuart Enterprises, Box 310, Grass Valley, CA 95945, is offering the ByteStalker Software Analyzer Card. The card includes 16 switches for selecting a breakpoint address, a three-position mode switch (run, stop, trap), and a two position step-hold switch. There are also six single hex digit displays, to show you address and contents. You can choose single-stepping at the machine cycle level, or the instruction level. The price is only \$88.50, plus postage and insurance. Contact J. Stuart at the above address or at phone (916) 273-9188.

We're Gaining on it...

We've gained about a week in our efforts to get AAL back on schedule: it looks like the April issue will actually get to the printer during April! Next month we'll be shooting to get the May issue in the mail before the end of May, and soon you'll be receiving each month's issue during the appropriate month. Thanks for your patience.

In the very first issue of AAL, back in October 1980, I wrote an article about an interpretive screen display subroutine. A lot of things have happened since 1980! Although perhaps it is insignificant compared to the rise and fall of nations, one change which dramatically affects Apple programmers is that now almost everything is done on an 80-column screen. That old message printer was tied to the 40-column screen, so I think an upgrade is in order.

The following program has more features than the 1980 version, and is also easier to use. Another nice change since 1980 is that most of the assemblers available for the Apple now include a macro capability. Macros make it easier to use special coded sequences of bytes like those necessary in the message printer.

Where is the message?

There are many different ways to tell a message-printing subroutine where the message to be printed is located in memory. The method I used in my 1980 article was to assign a message number to each message, and pass that number in the A-register; the message printer used the message number as an index to a table of message addresses. Sometimes I have used the message number technique without a table of message addresses: I put all the messages together in one consecutive area of memory; the message printer searched through the messages counting end-of-message characters until it found the N-th message, and then started printing.

I show two other techniques in the program which follows. The first one involves loading the full address of the message to be printed into the A- and Y-registers. In my program I put the high-byte of the address in the Y-register, and the low-byte in the A-register. Of course you could use any registers any way you wish, but this is what I used. The code for MESSAGE.PRINTER is shown in lines 1250-1380. I defined a macro named "MSG" in lines 1190-1230 to make it easier to write the code to call the message printer. Calls to the message printer may be written simply as one-line macro calls, such as:

```
>MSG TITLE  
>MSG COPYRIGHT
```

where TITLE and COPYRIGHT are the names of specific messages assembled elsewhere in a program.

The other technique I show here is probably the most popular one, if the programs I have disassembled are any indication. In this method the text of the message to be printed immediately follows the JSR to the message printer. The message printer uses the return address placed on the stack by the JSR to locate the message. After the message is printed, the return address is modified so that it points to the next instruction after the message before RTSing. ANOTHER.MESSAGE.PRINTER is shown in lines 1420-1640.

S-C Macro Assembler Version 2.0.....DOS \$100, ProDOS \$100, both for \$120	
Version 2.0 DOS Upgrade Kit for 1.0/1.1/1.2 owners.....\$20	
ProDOS Upgrade Kit for Version 2.0 DOS owners.....\$30	
Source Code of S-C Macro 2.0 (DOS or ProDOS).....each, additional \$100	
S-C Disassembler (ProDOS only) ...without source code \$30, with source \$50	
RAK-Ware DISASM (DOS only).....without source code \$30, with source \$50	
ProVIEW (ProDOS only).....\$20	
Full Screen Editor for S-C Macro (with complete source code).....\$49	
S-C Cross Reference Utility.....without source code \$20, with source \$50	
S-C Word Processor (with complete source code).....\$50	
DP18 and DPFP, including complete source and object code.....\$50	
ES-CAPE (Extended S-C Applesoft Program Editor).....	
Including Version 2.0 With Source Code.....\$50	
ES-CAPE Version 2.0 and Source Code Update (for Registered Owners).....\$30	
Bag of Tricks 2 (Quality Software).....(\$49.95) \$45 *	
MacASM -- Macro Assembler for Macintosh (Mainstay).....(\$150.00) \$50 *	
S-C Documentor (complete commented source code of Applesoft ROMs).....\$50	
Cross Assemblers for owners of S-C Macro Assembler.....\$32.50 to \$50 each	
(Available: 6800/1/2, 6301, 6805, 6809, 68000, Z-80, Z-8, 8048, 8051, 8085, 1802/4/5, PDP-11, GI1650/70, others)	
Beagle Bros. Applesoft Compiler (ProDOS only).....(\$74.95) \$65 *	
AAL Quarterly Disks.....each \$15, or any four for \$45	
Each disk contains the source code from three issues of AAL: Jan-Mar, Apr-Jun, Jul-Sep, and Oct-Dec.	
(All source code is formatted for S-C Macro Assembler. Other assemblers require some effort to convert file type and edit directives.)	
Diskettes (with hub rings)..... package of 25 for \$15 *	
Vinyl disk pages, 6"x8.5", hold two disks each.....10 for \$6 *	
Diskette Mailing Protectors (hold 1 or 2 disks).....40 cents each (Cardboard folders designed to fit 6"X9" Envelopes.) or \$25 per 100 *	
Envelopes for Diskette Mailers..... 6 cents each	
65802 Microprocessor (Western Design Center).....(\$95) \$50 *	
quikLoader EPROM System (SCRG).....(\$179) \$170 *	
PROMGRAMER (SCRG).....(\$149.50) \$140 *	
Minuteman 250 Uninterruptible Power Supply.....(\$359) \$320 +	
"Apple //gs Technical Reference", Fischer.....(\$19.95) \$19 *	
"65816/65802 Assembly Language Programming", Fischer.....(\$21.95) \$20 *	
"Programming the 65816", Eyes.....(\$22.95) \$21 *	
"Apple //e Reference Manual", Apple Computer.....(\$24.95) \$23 *	
"Apple //c Reference Manual", Apple Computer.....(\$24.95) \$23 *	
"ProDOS Technical Reference Manual", Apple Computer.....(\$29.95) \$27 *	
"Apple ProDOS: Advanced Features for Programmers", Little..(\$17.95) \$17 *	
"ProDOS Inside and Out", Doms & Weishaar.....(\$16.95) \$16 *	
"DOStalk Scrapbook", Weishaar & Kersey.....(\$14.95) \$14 *	
"Beneath Apple ProDOS", Worth & Lechner.....(\$19.95) \$18 *	
"Beneath Apple DOS", Worth & Lechner.....(\$19.95) \$18 *	
"Inside the Apple //c", Little.....(\$19.95) \$18 *	
"Inside the Apple //e", Little.....(\$19.95) \$18 *	
"Understanding the Apple //e", Sather.....(\$24.95) \$23 *	
"Understanding the Apple II", Sather.....(\$22.95) \$21 *	
"Apple II+//Ile Troubleshooting & Repair Guide", Brenner.....(\$19.95) \$18 *	
"Apple II Circuit Description", Gayler.....(\$22.95) \$21 *	
"Assem. Language for Applesoft Programmers", Finley & Myers.(\$18.95) \$18 *	
"Now That You Know Apple Assembly...", Gilder.....(\$19.95) \$18 *	
"Enhancing Your Apple II, vol. 1", Lancaster.....(\$15.95) \$15 *	
"Enhancing Your Apple II, vol. 2", Lancaster.....(\$17.95) \$17 *	
"Assembly Cookbook for the Apple II/Ile", Lancaster.....(\$21.95) \$20 *	
"Microcomputer Graphics", Myers.....(\$14.95) \$14 *	
"Assembly Lines -- the Book", Wagner.....(\$19.95) \$18 *	
"AppleVisions", Bishop & Grossberger.....(\$39.95) \$36 *	

* On these items add \$2.00 for the first item and
\$.75 for each additional item for US shipping.

+ Inquire for shipping cost.

 Foreign customers inquire for postage needed.

Texas residents please add 7.25 % sales tax to all orders.

*** S-C SOFTWARE, P. O. BOX 280300, Dallas, TX 75228 ***

*** (214) 324-2050 ***

*** Master Card, VISA, Discover and American Express ***

Both message printers begin by storing the address of the message into a page-zero pointer. Then a subroutine named GET.NEXT.CHAR in lines 1680-1750 is called to pick up characters of the message. Since the last instruction in this subroutine is the one which picks up the character, the Zero-status will be set if the \$00 byte indicating end-of-message is picked up.

What is in the Message?

The real intelligence of my message printers is in the subroutine PUT.CHAR, starting at line 1760. We have already mentioned that a \$00 byte indicates end-of-message. PUT.CHAR expects the X-register to contain a repeat count, and the A-register to contain the character to be printed. If the byte in the A-register has a value between \$40 and \$FF, it is sent to COUT to be printed (X) times. However, if the byte in the A-register has a value between \$01 and \$3F, it is interpreted in special way.

Values between \$80 and \$9F are control characters, and will be interpreted by the monitor firmware. If you are in an 80-column //e, //c, or //gs, the interpretation will be as shown in lines 1850-2070. In 40-column mode not all of the functions are active. A Viewmaster or other type of 80-column card will have its own set of controls.

Values between \$01 and \$18 are interpreted by PUT.CHAR as the first byte of a two-byte cursor position code. This first byte indicates the line number (1-24 decimal) to put the cursor on, and the following byte is the column number (0-79 decimal). I defined a macro "POS" in lines 2100-2130 to simplify putting position codes into messages.

Values between \$19 and \$3F are used to index a special function table. My table is shown in lines 2700-2840. Notice that I used a macro "FUN" to write each entry in the table. This macro defines the special function code as "F.name", and assembles the address-1 of the code which performs the function. I thought of seven handy functions, but you can probably think of a lot more. As you do, just add >FUN lines to the table and write subroutines to perform the functions. Use mine as models, and you should be successful.

The methods I used for setting up the cursor position and handling other special functions work with Apple's firmware in the 40- and 80-column modes; if you have a Viewmaster or other 80-column card, you will need to make some changes in PUT.CHAR and its subroutines.

I made macros for the all of the functions, shown in lines 2140-2460. You might also find it handy to define macros for some of the control-characters processed by the 80-column firmware, but I didn't do it here.

Even though I have the "POS" function for positioning the cursor on the screen, I still wanted the ability to do HTAB and VTAB independently. These are my first two functions. By

writing ">HTAB xx" you can position to column xx on the current line. ">VTAB yy" will put the cursor on line yy without changing the column. Again, yy is a number between 1 and 24, and xx runs from 0 to 79 (decimal).

If you remember, normal characters are printed N times according to the value in the X-register when you call PUT.CHAR. The REP function lets you set up the repeat count. ">REP n,char" will cause char to be printed n times.

The CEN funtion is perhaps the most powerful one. It centers a string on the current line. You simply write something like

```
>CEN "This string will be centered"
```

...and it will be. The code for this function is in lines 3020-3180. All characters which follow the CEN function code are counted, up to the next function code or control character. Then the position for the first character is computed by halving the difference between window width and string length, and an HTAB performed.

FIL and FCL will fill an entire line with a specified character. FIL lets you say which line, and FCL fills the current line. You could do the same thing by putting the cursor at the beginning of the line to be filled and then using a repeat count larger than the window width followed by the fill character, but this way is easier. Let the computer work for you!

WIN lets you specify a window on the screen. This is very rudimentary as window routines go, because all it does is load the four locations the monitor uses to define the window: left, width, top, and bottom. "Left" is the column number to start the window at, and is a value from 0 to 79. "Width" is the number of characters in a line, and has a value from 1 to 80. Just be sure that these two numbers do not add to a value greater than 80, unless you really want to clobber your program. "Top" is the line number of the top of the window, having a value from 0 to 23. "Bottom" is the line number of the bottom line of the window, plus 1; the value goes from 1 to 24. The full 80-column window would be set up with ">WIN 0,80,0,24".

How about some examples?

I wrote two sample programs. "T", in lines 3500-3540, calls MESSAGE.PRINTER to print the messages in lines 3560-3790. "TT", in lines 3950-4170, calls ANOTHER.MESSAGE.PRINTER to print embedded messages.

```
1010 *SAVE S-C.MSG.MACHINE
1020 *-----
20- 1030 MON.LEFT .EQ $20
21- 1040 MON.WIDTH .EQ $21
22- 1050 MON.TOP .EQ $22
23- 1060 MON.BOTTOM .EQ $23
24- 1070 MON.CH .EQ $24
25- 1080 MON.CV .EQ $25
1090 *-----
057B- 1100 MON.CH80 .EQ $57B
1110 *-----
```

DOS Anatomy

AN UNPRECEDENTED DISASSEMBLY

FEATURES:

- ONLY COMPLETE DISASSEMBLY of the DOS 3.3 operating system.
- entire operating system is described in **BOTH LINEAR AND FORMATTED FORMS.**
- all disassemblies are **HEAVILY COMMENTED.**
- describes numerous tips and techniques for modifying DOS and implementing sophisticated **PROTECTION SCHEMES.**
- describes a **LEGAL SOURCE OF FREE COMPUTER PAPER** that may be located in your area.
- Includes a **UNIQUE TEXT-FILE READING PROGRAM** (called TEXT READER) that can be used to read both random-access and sequential text files at binary speeds. This machine-language program is **LOADED WITH OPTIONS** and can be configured to accommodate various printers and interface cards. (Source code for the TEXT READER program is available at an additional cost).
- entire software package is **UNPROTECTED AND GUARANTEED.**
- high quality / low price: **64 FILES ON 15 DOUBLE-SIDED DISKS FOR ONLY \$100 (US Funds).**

DESIGNED FOR:

- software developers - computer magazine authors - protection artists
- computer hackers - instructors of assembly language
- any assembly language programmer who frequently foregoes sleep in an effort to reach new plateaus of understanding.

SYSTEM REQUIREMENTS:

- 1) Apple II+ (with 16K ram card), Apple IIe or Apple IIc.
- 2) One disk drive (using 5-1/4 inch diskettes).
- 3) DOS 3.3 operating system. (Note: In order to conserve disk space, all files are housed on DOS-less data disks. Therefore, you must boot with a normal version of DOS 3.3 before this software package can be used.)
- 4) Printer that can support a 132-character line length in condensed or normal modes. (A smaller printer width can be used but is not recommended.)

---Optional---

- 5) MERLIN (a.k.a. "BIG MAC") assembler. (An assembler is not required if you just wish to review the text file contents. Text files containing numerical and alphabetical cross-referenced symbol tables are provided to assist programmers who own a different brand of assembler.)
- 6) The book "BENEATH APPLE DOS" would prove helpful.

To order, send cheque or money order for \$100 (US Funds) to:

WINDSOR SCIENTIFIC LTD.
3621 - 109 Avenue Edmonton Alberta
Canada T5W 0G6
Phone (403) 479-2779

DOS 3.3, MERLIN and BENEATH APPLE DOS are respective trademarks or products of Apple Computer Inc., Roger Wagner Publishing Inc. and Quality Software.

FC22- 1120 MON.VTAB .EQ \$FC22
 FDDED- 1130 MON.COUT .EQ \$FDDED
 1140 *-----
 00- 1150 MSG.PNTR .EQ \$00,01
 1160 *-----
 1170 * Address of message in Y,A registers
 1180 *-----
 1190 .MA MSG >MSG <message address>
 1200 LDA #1
 1210 LDY /1
 1220 JSR MESSAGE.PRINTER
 1230 .EM
 1240 *-----
 1250 MESSAGE.PRINTER
 0800- 85 00 1260 STA MSG.PNTR Store message address
 0802- 84 01 1270 STY MSG.PNTR+1 into page-zero pointer
 0804- 8A 1280 TXA Save X-register
 0805- 48 1290 PHA
 0806- A2 01 1300 LDX #1 Clear repetition counter
 0808- 20 3F 08 1310 JSR GET.CHAR First char of message
 080B- F0 08 1320 BEQ .2 ..empty message!
 080D- 20 44 08 1330 .1 JSR PUT.CHAR Process the character
 0810- 20 39 08 1340 JSR GET.NEXT.CHAR get another one
 0813- D0 F8 1350 BNE .1 ...not at end of message yet
 0815- 68 1360 .2 PLA Restore X-register
 0816- AA 1370 TAX
 0817- 60 1380 RTS Return to caller
 1390 *-----
 1400 * Message follows JSR to message printer
 1410 *-----
 1420 ANOTHER.MESSAGE.PRINTER
 0818- 68 1430 PLA Store message address (-1)
 0819- 85 00 1440 STA MSG.PNTR into page-zero pointer
 081B- 68 1450 PLA
 081C- 85 01 1460 STA MSG.PNTR+1
 081E- 8A 1470 TXA Save X-register
 081F- 48 1480 PHA
 0820- 98 1490 TYA Save Y-register
 0821- 48 1500 PHA
 0822- A2 01 1510 LDX #1 Clear repetition counter
 0824- D0 03 1520 BNE .2 ..Always
 0826- 20 44 08 1530 .1 JSR PUT.CHAR Process a character
 0829- 20 39 08 1540 .2 JSR GET.NEXT.CHAR
 082C- D0 F8 1550 BNE .1 ...not at end of message yet
 082E- 68 1560 PLA Restore Y-register
 082F- A8 1570 TAX
 0830- 68 1580 PLA Restore X-register
 0831- AA 1590 TAX
 0832- A5 01 1600 LDA MSG.PNTR+1 Get return address back on stack
 0834- 48 1610 PHA
 0835- A5 00 1620 LDA MSG.PNTR
 0837- 48 1630 PHA
 0838- 60 1640 RTS Return to caller
 1650 *-----
 1660 * Get a message character
 1670 *-----
 1680 GET.NEXT.CHAR
 0839- E6 00 1690 INC MSG.PNTR
 083B- D0 02 1700 BNE GET.CHAR
 083D- E6 01 1710 INC MSG.PNTR+1
 1720 GET.CHAR
 083F- A0 00 1730 LDY #0
 0841- B1 00 1740 LDA (MSG.PNTR),Y
 0843- 60 1750 RTS
 1760 *-----
 1770 * Process a message character
 1780 *-----
 1790 * \$00 END OF MESSAGE
 1800 * \$YY XX where is between \$01 and \$18, position
 1810 * the cursor as vtab YY, htab XX.
 1820 * \$19-3F special functions
 1830 * \$40-FF print as is
 1840 *-----
 1850 * Characters between \$80 and \$9F are interpreted by
 1860 * the Apple 80-column firmware as follows:
 1870 *-----
 1880 * \$87 bell
 1890 * \$88 backspace
 1900 * \$8A move cursor down one line

```

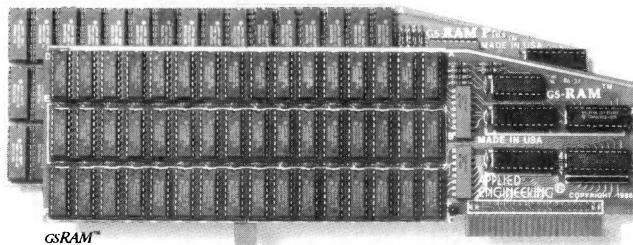
1910 *      $8B      clear from cursor to end of window
1920 *      $8C      clear window and home cursor
1930 *      $8D      carriage return
1940 *      $8E      normal video
1950 *      $8F      inverse video
1960 *      $91      switch to 40-column
1970 *      $92      switch to 80-column
1980 *      $95      de-activate firmware
1990 *      $96      scroll display down one line
2000 *      $97      scroll display up one line
2010 *      $98      disable Mouse Text
2020 *      $99      home cursor without clearing window
2030 *      $9A      clear current line
2040 *      $9B      enable Mouse Text
2050 *      $9C      move cursor forward one space
2060 *      $9D      clear to end of line
2070 *      $9F      move cursor up one line
2080 *
2090 *      Message Content Macros
2100 *
2110 .MA POS      Position cursor (vtab, htab)
2120 .DA #]1,#]2
2130 .EM
2140 *
2150 .MA REP      Repeat character N times
2160 .DA #F.REP,#]1,#"]2
2170 .EM
2180 *
2190 .MA CEN      Center a string on current line
2200 .DA #F.CEN
2210 .AS "-"]1"
2220 .EM
2230 *
2240 .MA STR      normal Ascii string
2250 .AS "-"]1"
2260 .EM
2270 *
2280 .MA FIL      Fill specified line with character
2290 .DA #F.FIL,#]1,#"]2
2300 .EM
2310 *
2320 .MA FCL      Fill current line with character
2330 .DA #F.FCL,#"]1
2340 .EM
2350 *
2360 .MA WIN      Define a new text window
2370 .DA #F.WIN,#]1,#]2,#]3,#]4
2380 .EM
2390 *
2400 .MA HTAB      Htab the cursor
2410 .DA #F.HTAB,#]1
2420 .EM
2430 *
2440 .MA VTAB      Vtab the cursor
2450 .DA #F.VTAB,#]1
2460 .EM
2470 *
2480 PUT.CHAR
0844- C9 40 2490      CMP #$40      IS IT PRINTABLE?
0846- 90 08 2500      BCC .2      ...NO, SPECIAL FUNCTION
0848- 20 ED FD 2510 .1      JSR MON.COUT
084B- CA 2520      DEX      Decrement repeat count
084C- D0 FA 2530      BNE .1      ...repeat the character
084E- E8 2540      INX      Put the repeat count back to 1
084F- 60 2550      RTS
2560 *----Process function char-----
0850- C9 19 2570 .2      CMP #$19      >POS VV,HH or special function?
0852- 90 1F 2580      BCC P.POS  ...>POS call
2590 *----Branch to function code-----
0854- 0A 2600      ASL      Double function # to make index
0855- A8 2610      TAY
0856- C0 42 2620      CPY #FUN.TBL.SZ
0858- B0 08 2630      BCS .3      ...Ignore illegal function
085A- B9 32 08 2640      LDA FUN.TBL+1,Y
085D- 48 2650      PHA      Put function code address-1 on stack
085E- B9 31 08 2660      LDA FUN.TBL,Y
0861- 48 2670      PHA
0862- 60 2680 .3      RTS      Go to it!
2690 *

```

Insist on GS RAM™ When You Buy Your IIgs™

Expand the IIgs RAM and ROM with the GS RAM or GS RAM Plus with ROM Pak. Available now with 256K to 8 MEG!

GS RAM Plus™



GS RAM™

Remember the 16K cards for the II+ and the 64K cards for the IIe? At the time, that much memory seemed like a lot. But when the owners of these memory cards came to us for more memory, many had to throw away their smaller Apple memory cards or try to sell them. Most of our customers told us that had they known about Applied Engineering's larger memory cards when they bought their Apple, they would have purchased them at the same time.

GS RAM and GS RAM Plus are available now, allowing up to 8 MEG of memory expansion. That's 8 times the memory capacity of Apple's card and just look at the benefits that only GS RAM and GS RAM Plus have over Apple's card:

- Lower cost
- Has 6 RAM banks (Apple's card has 4)
- Has memory expansion port
- Has ROM expansion port
- No configuration blocks to set
- No soldered in RAM chips
- Expandable to 8 MEG
- Expands AppleWorks internal limits
- Built-in Hi-Res self-diagnostic software
- 5 year hassle free warranty (Apple has a 90 day warranty)
- Made in USA

GS RAM for More AppleWorks Power

Only GS RAM and GS RAM Plus eliminate AppleWorks internal memory limits, increasing the maximum number of records available from 6,000 to over 25,000 and only GS RAM and GS RAM Plus increase the number of lines permitted in the word processing mode from 6,000 to over 15,000. And only GS RAM and GS RAM Plus offer a built-in printer buffer so you can continue using Appleworks while your printer is printing GS RAM and GS RAM Plus even expand the number of lines in the clipboard from 255 to 2047 and will auto segment large files so they can be saved on two or more disks. You can

even have Pinpoint or Macromworks and your favorite spelling checker in RAM for instant response. GS RAM and GS RAM Plus will even display the time and date right on the AppleWorks screen. Nothing comes close to enhancing AppleWorks so much.

Turn Your IIgs into a Giant

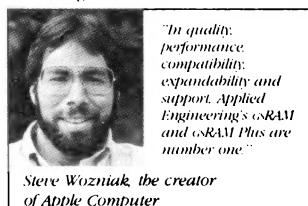
Simply plug GS RAM into the IIgs memory expansion slot and you've got up to 8 megabytes of RAM at your fingertips—all of it instantly and automatically recognized by the IIgs. GS RAM is compatible with all IIgs software, including AppleWorks, as well as BASIC®, ProDOS, DOS 3.3, PASCAL®, C and CP/M®.

Grow by Bytes or Megabytes

We offer GS RAM in two configurations so you can increase your memory 256K at a time (GS RAM) or a megabyte at a time (GS RAM Plus). Both offer full compatibility, lower cost than other boards, and easy expandability. And both are extremely low in power consumption. A fully expanded GS RAM operates at only 375 mA, and GS RAM Plus at only 270 mA (even with 6 megabytes on board!).

GS RAM—for Normal Memory Requirements

GS RAM is available with 256K, 512K, 1 MEG or 1.5 MEG of memory already on board. If you don't need the full 1.5 MEG now, you can choose a GS RAM with less memory and expand it up to 1.5 MEG in the future—or upgrade to GS RAM Plus for a small charge.



Steve Jobs, the creator of Apple Computer

"In quality, performance, compatibility, expandability and support, Applied Engineering's GS RAM and GS RAM Plus are number one."

With an optional piggyback card, you can expand GS RAM even higher than 1.5 MEG! (Other cards are only expandable to 1 MEG.)

GS RAM Plus—for Growing by Leaps and Bounds

GS RAM Plus is the first Apple memory card to use 1 MEG RAM chips on the main board. It's available with 1 to 6 MEG on board. If you don't need the whole 6 MEG now, you can buy a GS RAM Plus with less memory and easily expand it in the future.

GS RAM Plus can be expanded up to 8 MEG with an optional piggyback card.

Easy Expansion

Both GS RAM and GS RAM Plus use standard RAM chips that are readily available and just plug right in. So unlike other cards, you'll find expanding your GS RAM or GS RAM Plus easy, convenient and very economical. And with our optional ROM expansion module you can even increase the IIgs's ROM space and all in just one slot.

Quality and Support of the Industry Leader

Applied Engineering is the largest supplier of Apple peripherals in the world. We invented the first large RAM cards for the Apple. With a 5-year "no-hassle" warranty and outstanding technical support, you can be sure GS RAM and GS RAM Plus will deliver the performance you're looking for—or return them within 15 days for a full refund.

GS RAM with 256K	\$169
GS RAM with 512K	\$219
GS RAM with 1 MEG	\$299
GS RAM with 1.5 MEG	\$379
GS RAM with 2-8 MEG	CALL
GS RAM PLUS with 1 MEG	\$459
GS RAM PLUS with 2 MEG	\$759
GS RAM PLUS with 3-8 MEG	CALL

Order today!

See your dealer or call Applied Engineering today, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 6 1/4% sales tax. Add \$10.00 outside U.S.A.

AE APPLIED ENGINEERING™

The Apple enhancement experts.

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

CP/M is a registered trademark of Digital Research, Inc.

```

2700 .MA FUN
2710 F.]1 .EQ *-FUN.TBL/2
2720 .DA P.]1-1
2730 .EM
2740 *
2750 FUN.TBL .EQ *-$19-$19
2760 >FUN VTAB Vtab YY
2770 >FUN HTAB Htab XX
2780 >FUN REP Repeat, count, character
2790 >FUN CEN Center, string, null
2800 >FUN NUL ... ends centered string
2810 >FUN FIL Fill, line #, char
2820 >FUN FCL Fill current line, char
2830 >FUN WIN Window, left, width, top, bottom
2840 FUN.TBL.SZ .EQ *-FUN.TBL
2850 *
0831- 2750
0863- 2760
0865- 2770
0867- 2780
0869- 2790
086B- 2800
086D- 2810
086F- 2820
0871- 2830
42- 2840
0873- 20 BC 08 2850
0876- 20 39 08 2860 P.POS JSR MY.VTAB Do vertical tab
0879- C5 21 2870 P.HTAB JSR GET.NEXT.CHAR ...get horoz value
087B- 90 04 2880 CMP MON.WIDTH Stay inside current window
087D- A5 21 2890 BCC .1 ...it is inside
087F- E9 01 2900 LDA MON.WIDTH ...use right edge
0881- 85 24 2910 SBC #1
0883- 8D 7B 05 2920 .1 STA MON.CH 40-column horiz pos'n
0886- 60 2930 STA MON.CH80 80-column horiz pos'n
0887- 20 39 08 2940 RTS
088A- AA 2950 *
088B- 60 2960 P.REP JSR GET.NEXT.CHAR
088A- AA 2970 TAX REPETITION COUNT
088B- 60 2980 RTS
088C- A0 00 2990 *
088E- C8 3000 * Center a string on current line
088F- B1 00 3010 *
0891- 30 04 3020 P.CEN LDY #0 Count # chars in string
0893- C9 40 3030 .1 INY from next char to char below $40
0895- B0 F7 3040 LDA (MSG.PNTR) Y or btwn $80 and $9F
0897- C9 A0 3050 BMI .2 80-FF
0899- B0 F3 3060 CMP #$40
089B- 88 3070 BCS .1 40-7F, keep counting
089C- 98 3080 .2 CMP #$A0
089D- 49 FF 3090 BCS .1 A0-FF, keep counting
089F- 38 3100 DEY 00-3F or 80-9F, end of string
08A0- 65 21 3110 TYA (Line.width - string.len) / 2
08A2- 4A 3120 EOR #$FF gives starting horiz posn
08A3- 85 24 3130 SEC
08A5- 8D 7B 05 3140 ADC MON.WIDTH
08A8- 60 3150 LSR
08A9- 20 39 08 3160 STA MON.CH
08AC- 20 BC 08 3170 STA MON.CH80
08A8- 60 3180 P.NUL RTS
08A9- 20 39 08 3190 *
08AC- 20 BC 08 3200 P.FIL JSR GET.NEXT.CHAR
08AC- 20 BC 08 3210 JSR MY.VTAB
08AF- A9 00 3220 P.FCL
08B1- 85 24 3230 LDA #0
08B3- 8D 7B 05 3240 STA MON.CH
08B6- A6 21 3250 STA MON.CH80
08B8- 60 3260 LDX MON.WIDTH
08B9- 20 39 08 3270 RTS
08BC- C5 23 3280 *
08BE- 90 02 3290 P.VTAB JSR GET.NEXT.CHAR
08CO- A5 23 3300 MY.VTAB
08C2- 85 25 3310 CMP MON.BOTTOM Stay inside current window
08C4- C6 25 3320 BCC .1 ...it is inside
08C6- 4C 22 FC 3330 LDA MON.BOTTOM Go to bottom line
08C6- 4C 22 FC 3340 .1 STA MON.CV We use line # 1 to N, but
08C6- 4C 22 FC 3350 DEC MON.CV monitor uses 0 to N-1
08C6- 4C 22 FC 3360 JMP MON.VTAB
08C9- 20 39 08 3370 *
08CC- 85 20 3380 P.WIN
08CE- 20 39 08 3390 JSR GET.NEXT.CHAR
08D1- 85 21 3400 STA MON.LEFT
08D3- 20 39 08 3410 JSR GET.NEXT.CHAR
08D6- 85 22 3420 STA MON.WIDTH
08D8- 20 39 08 3430 JSR GET.NEXT.CHAR
08DB- 85 23 3440 STA MON.TOP
08DD- A9 99 3450 JSR GET.NEXT.CHAR
08DF- 4C ED FD 3460 STA MON.BOTTOM
08DF- 4C ED FD 3470 LDA #$99
08DF- 4C ED FD 3480 JMP MON.COUT

```

17 Meg Ile/1 Meg IIC

Want the best & easiest-to-use (most support) RAM card on the market — at the best price? Buy Checkmate Technology's State-Of-The-Art MULTIRAM RGB RAM CARD™ from us w/BATTERY BACKED-UP RAM options that save programs like AppleWorks for years! FAST CHEAP REPLACEMENT FOR HARD DISKS, USER EXPANDABLE TO 17 MEGS, (100%) compatible w/all 3rd party software/hardware, 65C816 option (\$147), sharp 80 columns/Double Hi-Res. & FREE RGB! Direct substitute for Ramworks I-III™ (we think it's better) by Applied Engineering (also known as Preferred Computing) or Apple 80 col cards, has a 5 YR WARRANTY, 15 day money back guarantee, & FREE AUTOMATIC SOFTWARE UPDATES ONLY FROM US!

MultiRam RGB expands to 1 Meg + optional BATTERY PROTECTABLE MultiRam Plus piggyback expandable to 16 Meg's. MultiRam Ile expands to 768k. The software & manual is the best & easiest-to-use on the market — without exception!

FREE APPLEWORKS EXPANDER SOFTWARE W/EVERY CARD — loads ALL versions of Appleworks (in 30 seconds w/ AW 2.0), even printer routines, runs faster, Desktop 2048k, 23,000+ records & 4,800+ word procilines, segments/lgs files onto multiple disks! **FREE AW/TIME/DAY/DATE ON-SCREEN w/ clock, AUTO-COPY TO RAM, PRODOS/DOS 3.3, RAM DISK & TEST.** Optional CP/M & Pascal Ram disks (\$20/ea)!

	MultiRam	MultiRam
	RGB Card	Ile Card
64k MultiRam	159.	129.
320k MultiRam	189.	157.
576k MultiRam	219.	187.
832k MultiRam	244.	212.
1024k MultiRam	269.	↔ ↔ ↔
MultiRam		
Plus Piggyback		
256k MultiRam Plus	219.	NEW
512k MultiRam Plus	269.	NEW
768k MultiRam Plus	316.	NEW
1024k MultiRam Plus	359.	NEW
2-17 meg MultiRam Plus	Call	NEW
Battery/AC Kit (for Plus)	36.	NEW

WHY BUY FROM COIT VALLEY COMPUTERS RATHER THAN SOME MAIL ORDER HOUSES? Only we offer a 15 day MultiRam complete money back satisfaction guarantee, extra software, more support, free automatic software updates, free 64k w/ each 256k/512k/768k Ile/RGB card. We know the products, & stock them for fast shipping! CALL FOR SUPPORT, QUESTIONS, PRICES, DISCOUNTS. SCHOOLS & GROUPS WELCOME.

COIT VALLEY COMPUTERS
(214) 234-5047

MCI 2969684
Tele 6502969684 MCI UW

Checkmate Technology's MultiRam CX card easily expands your IIC to 640k and has a CX+ Piggyback to add another 512k (1152k total)! It's 100% compatible with all IIC software/hardware & upgradable with a 65C816 kit (\$95) — all at about 50% less power than Z-RAM™! Has the **SAME FREE SOFTWARE/UPDATES & 5 YR WARRANTY AS Ile CARDS.** Buy a C-VUE flat panel display for \$287 w/ 512k or greater card.

256k MULTIRAM CX	155.
512k MULTIRAM CX	184.
768k MULTIRAM CX Combo	328.
1024k MULTIRAM CX Combo	354.
256k MULTIRAM CX+	199.
512k MULTIRAM CX+	209.
1200 Baud IIC Modem (add \$20 for cable)	147.
Clip-on IIC Rechargeable Battery	115.
IIC Flat Panel Display (w/ 512k > Mram)	287.
IIC System Clock	69.

Cermek 1200 internal modem Ile/Ii+/Iigs . 148.
(+ FREE Source subscription/ VT 100 Shareware/ ASCII Express & ProTERM™ Setup Instructions)

Slot 1-7 Ram Card for II+, Ile, Laser	Call
ProAPP 20 Meg Hard Disks (+ FREE software)	Call
Apple Ile Enhancement Kit	62.
Apple Iigs	Call
Apple 3.5 800k Unidrive (Ile/IIC/Iigs)	319.
Super Serial Compatible Card	87.
RamPak 512k gs RAM card	196.
Apple Compatible Drive Ile/II+ (Iigs/Ile + \$19)	132.
Clockworks Clock (Thunder/TimeHO™ compat)	84.
13" RGB Monitor w/ cables (req. RGB card)	327.
Heavy Duty Power Supply II/II+/Ile	69.
Ile Programmable Numeric Keypad	85.
System Saver Fan (II/II+/Ile)	66.
Uninterruptable Power Supply 250 watt	299.
AutoWorks (Best AW Macro + more)	35.
FontWorks (Best AW sideways/Fonts & more)	35.
Pinpoint Program	55.
Pinpoint Document Checker (Best Speller)	45.
Copy II Plus 7.3 (Best ProDOS Utility)	29.
ProTERM Modem Software	84.
ASCII Express Modem Software	84.
VIP Professional (Lotus Clone. Specify Comp)	169.

Terms: Add \$4- Ground or \$6-Air shipping & phone # to each U.S. MultiRam card order (foreign orders/FPO/APO extra). Add 3% for MasterCard/Visa/Am Exp (include expir date) & P.O.'s. For fast delivery send Cashier's/Certified check/Money Order C.O.D. (add \$5) & personal checks accepted (allow 16 days). Tax res add 7 1/4% tax. Software non-returnable.

Ordering: PRINT Name & address, Res & Bus phone, credit card name/number/expir date where appropriate, qty/description/price/shipping charges if known. Call for questions.

Autoworks, Ramworks II/II/II+, Master II HO/Z ram, respective trademarks of Software Touch, Applied Engineering. Prices subject to change

14055 Waterfall Way
Dallas, Texas 75240

```

08E2- 3490 *-----#
08E9- 3500 T >MSG MSG0
08F0- 3510 >MSG MSG2
08F7- 3520 >MSG MSG1
08FE- 3530 >MSG MSG3
08FE- 60 3540 RTS
08FF- 3550 *-----#
08FF- 8C 3560 MSG0 .HS 8C HOME SCREEN
0900- 3570 >WIN 10,38,0,24
0905- 3580 >FIL 1,- FILL TOP LINE WITH DASHES
0908- FC 3590 .HS FC ONE BAR ON NEXT LINE
0909- 8D 88 FC 3600 .HS 8D.88.FC.FC 2 BARS
090D- 8D 88 FC 3610 .HS 8D.88.FC.FC 2 BARS
0910- FC 3620 .HS 8D.88.FC.FC 2 BARS
0911- 8D 88 FC 3630 .HS 8D.88.FC.FC 2 BARS
0914- FC 3640 .HS 8D.88.FC.FC 2 BARS
0915- 8D 88 FC 3650 .HS 8D.88.FC.FC 2 BARS
0919- 8D 88 FC 3660 .HS 8D.88.FC 1 BAR AT END OF LINE
0921- 8D 88 FC 3670 >FCL - FULL CURRENT LINE WITH DASHES
0924- 3680 >POS 3,0 VTAB 3, HTAB 0
0926- 3690 .HS 8F INVERSE MODE
0928- 8F 3700 >CEN "Demonstration of Message Printer"
094A- 8E 8D 8D 3710 .HS 8E.8D.8D Normal mode
094D- 3720 >CEN "S-C Software Corporation"
0966- 8D 3730 .HS 8D
0967- 8D 3740 >CEN "P. O. Box 280300"
0978- 8D 3750 .HS 8D
0979- 3760 >CEN "Dallas, TX 75228"
098A- 8D 8D 8D 3770 .HS 8D.8D.8D
098D- 3780 >WIN 0,80,0,24 Full screen again
0992- 00 3790 .HS 00 End of message
0993- 3800 *-----#
0995- 9D 3810 MSG1 >POS 11,1 VTAB 11, HTAB 1
0996- 3820 .HS 9D CLR EOL
09A3- 00 3830 >STR "SELECT ONE: "
09A4- 3840 .HS 00 End of message
09A6- 8F 3850 *-----#
09A7- 3860 MSG2 >POS 24,1 VTAB 24, HTAB 1
09A7- 3870 .HS 8F INVERSE MODE
09CC- 8E 00 3880 >STR "<SPACE> FOR MENU, <RETURN> FOR MORE "
0990- 3890 .HS 8E.00 NORMAL MODE, ECM
09CE- 87 87 8D 3910 MSG3 .HS 87.87.8D
09D1- 3920 >STR "Wasn't that a nice demonstration?"
09F2- 8D 00 3930 .HS 8D.00
09F4- 20 18 08 3940 *-----#
09F7- 8D 8D 3950 TT JSR ANOTHER.MESSAGE.PRINTER
09F9- 3960 .HS 8D.8D
09A20- 3970 >CEN "This is another way to print messages."
09A22- 3980 >HTAB 0
09A25- 3990 >STR "<<<" TAB TO END OF LINE
09A27- 88 88 4000 >HTAB 99
09A29- 4010 .HS 88.88 BACK UP TWO SPACES
09A2C- 8D 00 4020 >STR ">>>"
09A2E- 20 18 08 4030 .HS 8D.00
09A31- 4040 JSR ANOTHER.MESSAGE.PRINTER
09A34- 4050 >REP 10,M-$40 10 CARRIAGE RETURNS
09A39- 4060 >WIN 0,1,14,24
09A43C- 4070 >REP 10,1
09A41- 4080 >WIN 79,1,14,24
09A44- 4090 >REP 10
09A49- 4100 >WIN 0,80,13,24
09A4B- 4110 >FCL
09A4E- 4120 >FIL 23,-
09A53- 4130 >WIN 0,80,0,24
09A55- 00 4140 >VTAB 24
09A56- 60 4150 .HS 00 End of message
09A56- 4160 RTS
4170 *-----#

```

Since we published my patches to allow DOS 3.3 to use the UniDisk 3.5 last May and June I've been using it every day, and just loving all that space. After basking in readers' praise for such a useful piece of code, you can imagine how I felt when I first went to boot that disk on the IIgs and watched it crash into the dirt!

After a little detective work the cause is found: When we boot from the //e Protocol Converter Interface it passes slot*16 in the A- and X-registers and the slot number in location \$58. In my boot code in FORMAT.UNIDISK I was using the A-register and the contents of \$58 to set up my RWTS. Come to find out, the IIgs only gives us the slot*16 in the X-register, so we have to do a little more manipulation to create the values we need. Here is the revised code:

3080	BOOT	TXA	
3082		PHA	save slot*16
3090		LSR	.divide
3092		LSR	..by 16
3094		LSR	...to get
3096		LSRslot

To update the boot image on an existing disk you can use the instructions at the top of page 23 of the July '86 issue. Be sure to note the caution in the last paragraph of page 21 in that issue, about not installing a DOS image that already contains RAMdisk patches: I tripped over that again myself!

We Make Measurement And Control Easy!

12 BIT, 16 CHANNEL, PROGRAMMABLE GAIN A/D

- All new 1984 design incorporates the latest in state-of-art I.C. technologies
- Complete 12 bit A/D converter, with an accuracy of 0.02%
- 16 single ended channels (single ended means that your signals are measured against the Apple's GND) or 8 differential channels. Most all the 16 channels you will measure are single ended.
- 9 software programmable full scale ranges, any of the 16 channels can have any range at any time. Under program control, you can select any of the following ranges: ±10 volts, ±5V, ±2.5V, ±1.0V, ±500MV, ±250MV, ±100MV, ±50MV or ±25MV.
- Very fast conversion (25 micro seconds).
- Analog input resistance greater than 1,000,000 ohms.
- Laser-trimmed scaling resistors.
- Low power consumption through the use of CMOS devices.
- The user connector has +12 and -12 volts on it so you can power your sensors.
- Only elementary programming is required to use the A/D.
- The entire system is on one standard 160 pin plug in card that fits neatly inside the Apple.
- System includes sample programs on disk.

PRICE \$319

A few applications may include the motion timer • flow • temperature • humidity • wind speed • wind direction • light intensity • pressure • RPM • soil moisture and many more

PRICE \$319

A/D & D/A

- A/D & D/A Features:
 - Single PC card
 - 8 channels A/D
 - 8 channels D/A
 - Very fast conversion time
 - Many analog ranges
 - Manual contains sample applications

A/D SPECIFICATIONS

- 0.3% accuracy
- On-board memory
- Fast conversion (0.78 MS per channel)
- A/D process totally transparent to Apple (looks like memory)
- User programmable input ranges are ±10 to ±2.5 to ±1.0 to ±0.5 to ±0.25 to ±0.125 to ±0.0625
- The A/D process takes place on a continuous, channel sequencing basis. Data is automatically transferred to its proper location in the on-board RAM. No A/D converter could be easier to use.

D/A SPECIFICATIONS

- 0.3% accuracy
- On-board memory
- On-board output buffer amps can drive 10 MA
- D/A process is totally transparent to the Apple (just poke the data)
- Fast conversion (0.03 MS per channel)
- User programmable output ranges are 0 to 5 volts and 0 to 10 volts
- The D/A section contains 8 digital to analog converters with output buffer amplifiers and interface to on-board memory. All data bits are provided for each of the eight D/A converters. No D/A converter could be easier to use. The on-board amplifiers are laser-trimmed during manufacture, thereby eliminating any requirement for off-set nulling

PRICE \$199

SIGNAL CONDITIONER

Our 8 channel signal conditioner is designed for use with both our A/D converters. This board incorporates 8 FET op-amps, which allow almost any gain or offset. For example, an input signal that varies from 2.00 to 2.15 volts or a signal that varies from 0 to 50 mV can easily be converted to 0-10V output for the A/D.

The signal conditioner's outputs are on a high-quality 16 pin gold I.C. socket that matches the one on the A/D's so a simple ribbon cable connects the two. The signal conditioner can be powered by your Apple or from an external supply.

FEATURES

- 4" square for standard card cage and 4 mounting holes for standard mounting. The signal conditioner does not plug into the Apple. It can be located up to ½ mile away from the A/D.
- 22 pin 156 spacing edge card input connector (extra connectors are easily available i.e. Radio Shack)
- Large bread board area.
- Full detailed schematic included.

PRICE \$79

I/O 32

- Provides 4, 8-Bit programmable I/O Ports
- Any of the 4 ports can be programmed as an input or an output port
- All I/O lines are TTL (0-5 volt) compatible
- Your inputs can be anything from high speed logic to simple switches
- Programming is made very easy by powerful on-board firmware
- The I/O 32 is your best choice for any control application

The I/O manual includes many programs for inputs and outputs.

Some applications include:

Burglar alarm, direction sensing, use with relays to turn on lights, sound buzzers, start motors, control tape recorders and printers, use with digital joystick

PRICE \$89

Please see our other full page ad in this magazine for information on Applied Engineering's Timemaster Clock Card and other products for the Apple. Our boards are far superior to most of the consumer electronics made today. All I.C.'s are in high quality sockets with mil-spec components used throughout. PC boards are glass-epoxy with gold contacts. Made in America to be the best in the world. All products compatible with Apple II and IIe.

Applied Engineering's products are fully tested and available for immediate delivery. All products are guaranteed with a no hassle three year warranty.

Texas Residents Add 5% Sales Tax
Add \$10.00 If Outside U.S.A.

Send Check or Money Order to
APPLIED ENGINEERING
P.O. Box 798
Carrollton, TX 75006

Call (214)241-4060
9 a.m. to 11 p.m. 7 days a week
MasterCard, Visa & C.O.D. Welcome
No extra charge for credit cards

Digging into the 65816 Apple IIgs has provided more entertainment than I've had since the introduction of the Apple II. So far, my programming efforts have been in the ProDOS 8 environment. When Apple gets its act together on ProDOS 16, writing position-independent code will become critical. Why not do a bit of preparing now? (The following techniques will apply to older Apples which have been upgraded with a 65802 cpu chip, also.)

Absolute internal references are the bugaboo of the position-independent programmer. (By the way, I am not referring to those individuals who program as well standing on their heads as they do flat on their backs or seated in a chair; I am referring to programs which run correctly no matter where they are loaded into memory, without any modifications to reflect the position in memory.) The three most common absolute internal references are:

- (1) the JMP to a program location;
- (2) the JSR to a subroutine location; and
- (3) the reading from or writing to program text or data tables or variables.

The first problem can be solved simply by using the BRL instruction instead of JMP. For avoiding the second and third traps, the PER opcode is an ideal tool. The solutions outlined below pertain only to code and data within one 64K bank of memory. Solutions which handle programs and data in multiple banks will have to wait for future articles.

PER is one of the new opcodes found in the 65802 and 65816. PER stands for "Push Effective program counter (PC) Relative data word onto stack", and employs a hybrid stack/PC-relative addressing mode. PER adds the 16-bit signed displacement in its two-byte operand to the current value for the PC and pushes the result onto the stack. As usual, the high byte is pushed first, then the low byte. Regardless of the values for the m-, x-, and e-status bits, two bytes are always pushed onto the stack. Because the operand is a displacement relative to the PC, the top two bytes of the stack always will point to the designated program code or data, irrespective of whether it is internal or external to the program. Thus, an internal JSR can be simulated by the following sequence:

```
PER CONTINUE-1
BRA SUBROUTINE
CONTINUE ...
...
SUBROUTINE ...
...
RTS
```

Note that the continuation address to be placed on the stack must be one less than the desired address (CONTINUE-1 above),

because the RTS opcode increments the address it finds on the stack before jumping to that address. Also note that the BRA opcode works well if the subroutine is nearby (within about + or - 128 bytes of the BRA itself); if the subroutine is further away, you will need to use the BRL opcode here.

I have constructed two macros in the listing of the program which follows. >BSR calls nearby subroutines, and >BSL calls both nearby and distant subroutines. The macro definitions are in lines 1130-1210, and I have shown them in use in lines 1320-1330. The listed program itself is rather trivial, designed just to illustrate my contrived solutions. Lines 1270-1340 clear the screen and then print out the hex values from \$00 to \$F0, with an interval of \$10. The two loops in lines 1350-1560 show two different ways to access data which is part of a program and still be position-independent; their function is to print out a string.

To access internal data, PER again comes to the rescue by allowing us to establish an external (zero page or stack) pointer to the data. Here, the operand must represent the exact address of the target data. The first example stores the pointer generated by PER into a pointer-variable in page zero (lines 1360-1400). PER puts the pointer onto the stack, and two PLAs pull it back off. If we were in Native mode with the m-bit cleared for 16-bit operation, only one PLA and STA would be needed. We could replace lines 1370-1400 with:

CLC	
XCE	ENTER NATIVE MODE
REP #\$20	M=0, 16-BIT DATA
PLA	PULL OFF BOTH BYTES
STA PTR	STORE BOTH BYTES
XCE	BACK TO EMULATION MODE

The second approach is to leave the REP-generated pointer on the stack. Line 1500 shows how we can use it with stack-relative indirect-indexed address mode to access the same text string. Lines 1550 and 1560 are then absolutely necessary, to remove the pointer from the stack when we are finished with it.

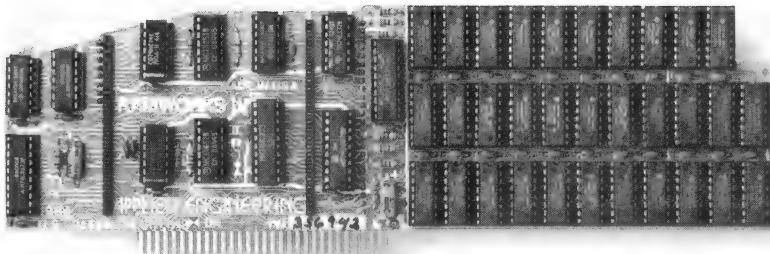
Though it is not appropriate for this sample program, the power of native mode is evident when working with 16-bit address pointers. The problem of using native mode in simple examples is that you cannot call the old standard monitor entry points (such as HOME, COUT, and PRBYTE) from native mode.

If you are going to write code which will be crossing 64K banks, you can still use the PER instruction. However, you also have to take into account the program bank and data bank registers. More on this at another time.

```
1000 *SAVE S.POSITION
1010 -----
1020 * POSITION INDEPENDENT CODE WITH THE 65802/16
1030 -----
1040 .OP 65816
1050 -----
1060 NUM .EQ $06
1070 PTR .EQ $07,08
1080 -----
```

RamWorks® III

Patented Performance from the Recognized Leader



With battery backed RAM port, RGB port, increased memory capacity, full software compatibility and more compact design, RamWorks III is a generation ahead.

RamWorks III is the newest 3rd generation RAM card for the Apple IIe. It incorporates all of the technology and improvements that years of experience and over a hundred thousand sales have given us. By selling more memory cards than anyone else and listening to our customers, we were able to design a memory card that has the ultimate in performance, quality, compatibility and ease of use. A design so advanced it's patented. We call it RamWorks III, you'll call it awesome!

The AppleWorks Amplifier.

While RamWorks III is recognized by all memory intensive programs, NO other expansion card comes close to offering the multitude of enhancements to AppleWorks that RamWorks III does. Naturally, you'd expect RamWorks III to expand the available desktop, after all Applied Engineering was a year ahead of everyone else *including Apple* in offering more than 55K, and we still provide the largest AppleWorks desktops available. But a larger desktop is just part of the story. Look at all the AppleWorks enhancements that even Apple's own card does not provide and *only* RamWorks III does. With a 256K or larger RamWorks III, *all* of AppleWorks (including printer routines) will automatically load itself into RAM dramatically increasing speed by eliminating the time required to access the program disk drive. Switch from word processing to spreadsheet to database at the speed of light with no wear on disk drives.

Only RamWorks eliminates AppleWorks' internal memory limits, increasing the maximum number of records available from 1,350 to over 25,000. *Only* RamWorks increases the number of lines permitted in the word processing mode from 2,250 to over 15,000. And *only* RamWorks offers a built-in printer buffer, so you won't have to wait for your printer to stop before returning to AppleWorks. RamWorks even expands the clipboard. And auto segments large files so they can be saved on two or more disks. You can even have Pinpoint or MacroWorks and your favorite spelling checker in RAM for instant response.

RamWorks, *nothing* comes close to enhancing AppleWorks so much.

The Most Friendly, Most Compatible Card Available.

Using RamWorks III couldn't be easier because it's compatible with more off-the-shelf software than any other RAM card. Popular programs like AppleWorks, Pinpoint, Catalyst, MouseDesk, HowardSoft, FlashCalc, Pro-File, Managing Your Money, SuperCalc 3a, and MagiCalc to name a few (and *all* hardware add on's like ProFile and Sider hard disks). RamWorks is even compatible with software written for Apple cards. But unlike other cards, RamWorks plugs into the IIe auxiliary slot providing our super sharp 80 column text (U.S. Patent #4601081) in a completely integrated system while leaving expansion slots 1 through 7 available for other peripheral cards.

RamWorks III is compatible with all

Apple IIe's, enhanced, unenhanced, American or European versions.

Highest Memory Expansion.

Applied Engineering has always offered the largest memory for the IIe and RamWorks III continues that tradition by expanding to 1 full MEG on the main card using standard RAMs, more than most will ever need (1 meg is about 500 pages of text)...but if you do ever need more than 1 MEG, RamWorks III has the widest selection of expander cards available. Additional 512K, 2 MEG, or 16 MEG cards just snap directly onto RamWorks III by plugging into the industry's only low profile (no slot 1 interference) fully decoded memory expansion connector. You can also choose non-volatile, power independent expanders allowing permanent storage for up to 20 years.

It Even Corrects Mistakes.

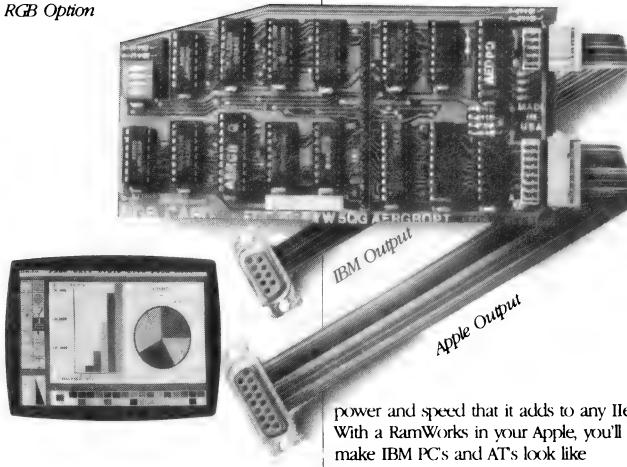
If you've got some other RAM card that's not being recognized by your programs, and you want RamWorks III, you're in luck. Because all you have to do is plug the memory chips from your current card into the expansion sockets on RamWorks to recapture most of your investment!

The Ultimate in RGB Color.

RGB color is an option on RamWorks and with good reason. Some others combine RGB color output with their memory cards, but that's unfair for those who don't need RGB *and* for those that do. Because if you don't need RGB

Applied Engineering doesn't make you buy it, but if you want RGB output you're in for a nice surprise because the RamWorks RGB option offers better color graphics plus a more readable 80 column text (that blows away any composite color monitor). For only \$129 it can be added to RamWorks giving you a razor sharp, vivid brilliance that most claim is the best they have ever seen. You'll also appreciate the multiple text colors (others only have green) that come standard. But the RamWorks RGB option is more than just the ultimate in color output because unlike others, it's fully compatible with all the Apple standards for RGB output control, making it more compatible with off-the-shelf software. With its FCC certified design, you can use almost any RGB monitor because only the new RamWorks RGB option provides both Apple standard and IBM standard RGB outputs (cables included). The RGB option plugs into the back of RamWorks with no slot 1 interference.

RGB Option



ference and remember you can order the RGB option with your RamWorks or add it on at a later date.

True 65C816 16 Bit Power.

RamWorks III has a built-in 65C816 CPU port for direct connection to our optional 65C816 card. The only one capable of linearly addressing more than 1 meg of memory for power applications like running the Lotus 1-2-3™ compatible program, VIP Professional. Our 65C816 card does not use another slot but replaces the 65C02 yet maintains full 8 bit compatibility.

Endorsed by the Experts.

A+ magazine said "Applied Engineering's RamWorks is a boon to those who must use large files with AppleWorks...I like the product so much that I am buying one for my own system." *inCider* magazine said "RamWorks is the most



"I wanted a memory card for my Apple that was fast, easy to use, and very compatible; so I bought RamWorks"

Steve Jobs, the creator of Apple Computer

powerful auxiliary slot memory card available for your IIe, and I rate it four stars. For my money, Applied Engineering's RamWorks is king of the hill."

Apple experts everywhere are impressed by RamWorks's expandability, versatility, ease of use, and the sheer

power and speed that it adds to any IIe. With a RamWorks in your Apple, you'll make IBM PC's and AT's look like slowpokes.

Quality and Support of the Industry Leader

RamWorks III is from Applied Engineering, the largest, most well supported manufacturer of Apple peripherals and the inventor of large RAM cards for the Apple. With our 5 year no hassle warranty and outstanding technical support, you're assured of the most trouble free product you can buy.

It's Got It All

- 15 day money back guarantee
- 5 year hassle free warranty insures

coverage no matter where you purchase

- Built-in super sharp 80 column display, (U.S. Patent #4601081)
- Expandable to 1 MEG on main card
- Expandable to 16 meg with expander cards, with NO slot 1 interference
- Can use 64K or 256K RAMs
- Powerful linear addressing 16 bit coprocessor port
- Automatic AppleWorks expansion up to 3017K desktop
- Accelerates AppleWorks
- Built-in AppleWorks printer buffer
- The only large RAM card that's 100% compatible with all IIe software
- RamDrive™ the ultimate disk emulation software included free
- Memory is easily partitioned allowing many programs in memory at once
- Compatible, RGB option featuring ultra high resolution color graphics and multiple text colors, with cables for both Apple and IBM type monitors
- Built-in self diagnostics software
- Lowest power consumption (U.S. Patent #4601081)
- Takes only one slot (auxiliary) even when fully expanded
- Socketed and user upgradeable
- Software industry standard
- Advanced Computer Aided Design
- Used by Apple Computer, Steve Jobs and virtually all software companies
- Displays date and time on the AppleWorks screen with any PRO-DOS compatible clock
- Much, much more!

RamWorks III with 64K	\$179
RamWorks III with 256K	\$219
RamWorks III with 512K	\$269
RamWorks III with 1 MEG	\$369
RamWorks III with 1.5 MEG	\$539
RamWorks III with 2 to 16 MEG	CALL
65C816 16 BIT Card	\$159
RGB Option	\$129
Optional Software:	
Pinpoint with RAM	
Enhancement Software	\$79
VIP Professional	\$219

RamWorks III. The industry standard for memory expansion of the Apple IIe.

ORDER YOUR RamWorks III TODAY. 9 a.m. to 11 p.m. 7 days, or send check or money order to Applied Engineering.

MasterCard, Visa and C.O.D. welcome. Texas residents add 5 1/2% sales tax. Add \$10.00 if outside U.S.A.

A Applied Engineering™
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060

FC58- 1090 HOME .EQ \$FC58
 FDDA- 1100 PRBYTE .EQ \$FDDA
 FDDE- 1110 COUT .EQ \$FDED
 1120 *-----
 1130 .MA BSR BRANCH RELATIVE TO SUBROUTINE
 1140 PER :1+1 When Subroutine is Nearby
 1150 :1 BRA j1
 1160 .EM
 1170 *-----
 1180 .MA BSL BRANCH RELATIVE TO SUBROUTINE
 1190 PER :1+2 When Subroutine is Far Away
 1200 :1 BRL j1
 1210 .EM
 1220 *-----
 1230 TRIVIAL EXAMPLE
 000800- 38 1240 SEC SET EMULATION MODE
 000801- FB 1250 XCE IN CASE CALLED FROM NATIVE MODE
 000802- 08 1260 PHP SAVE E-BIT
 1270 *---Show Position-Independent subroutine calls-----
 000803- 20 58 FC 1280 JSR HOME CLEAR SCREEN
 000806- A9 00 1290 LDA #0 FOR NUM = 0 TO \$F0 STEP \$10
 .000808- 85 06 1300 .1 STA NUM
 00080A- 20 DA FD 1310 JSR PRBYTE Print in hexadecimal
 00080D- 1320 >BSL PRBLNK Print two blanks
 00080D- 62 02 00 0000> PER :1+2 When Subroutine is Far Away
 000810- 82 36 01 0000> :1 BRL PRBLNK
 000813- 1330 >BSR ADD.10 ...NEXT NUM
 000813- 62 01 00 0000> PER :1+1 When Subroutine is Nearby
 000816- 80 2B 0000> :1 BRA ADD.10
 000818- 90 EE 1340 BCC .1 ...NOT FINISHED YET
 1350 *---Show Page-Zero Indirect data access-----
 00081A- 62 34 01 1360 PER TEXT PUT POINTER TO TEXT ON STACK
 00081D- 68 1370 PLA Now put pointer into page zero
 00081E- 85 07 1380 STA PTR
 000820- 68 1390 PLA
 000821- 85 08 1400 STA PTR+1
 000823- A0 00 1410 LDY #0 Point to beginning of text string
 000825- B1 07 1420 .2 LDA (PTR),Y Get next character of string
 000827- F0 06 1430 BEQ .3 End of string
 000829- 20 ED FD 1440 JSR COUT
 00082C- C8 1450 INY
 00082D- DO F6 1460 BNE .2 ...ALWAYS
 1470 *---Show Stack-Indirect data access-----
 00082F- 62 1F 01 1480 .3 PER TEXT POINTER TO TEXT ON STACK
 000832- A0 00 1490 LDY #0 Point to beginning of text string
 000834- B3 01 1500 .4 LDA (1,S),Y Get next character of string
 000836- F0 06 1510 BEQ .5 End of string
 000838- 20 ED FD 1520 JSR COUT
 00083B- C8 1530 INY
 00083C- DO F6 1540 BNE .4 ...ALWAYS
 00083E- 68 1550 .5 PLA POP OFF TEXT POINTER
 00083F- 68 1560 PLA
 1570 *---Return to caller-----
 000840- 28 1580 PLP GET SAVED E-BIT
 000841- FB 1590 XCE RESTORE E
 000842- 60 1600 RTS
 1610 *-----
 000843- 18 1620 ADD.10 CLC
 000844- A5 06 1630 LDA NUM ADD \$10 TO NUM
 000846- 69 10 1640 ADC #\$10 SET CARRY IF WRAPS AROUND
 000848- 60 1650 RTS
 1660 *-----
 000849- 1670 .BS 256 FILLER TO MOVE PRBLNK FURTHER AWAY
 1680 *-----
 000949- A9 A0 1690 PRBLNK LDA #"
 00094B- 20 ED FD 1700 JSR COUT
 00094E- 4C ED FD 1710 JMP COUT
 1720 *-----
 000951- 8D 1730 TEXT .HS 8D
 000952- DO EF F3 E9
 000956- F4 E9 EF EE
 00095A- AD C9 EE E4
 00095E- E5 F0 E5 EE
 000962- E4 E5 EE F4
 000966- A0 D4 E5 F8
 00096A- F4 A0 D3 F4
 00096E- F2 E9 EE E7 1740 .AS -/Position-Independent Text String/
 000972- 00 1750 .HS 00
 1760 *-----

Simple Line-Input Subroutine.....Bob Sander-Cederlof

The other day I needed a simple subroutine to read a line of text into an assembly language program. The routine had to be short, work in both 40- and 80-columns, and allow very simple editing. It had to read directly from the keyboard, regardless of where the input vector at KSWL (\$38,39) might be pointing.

The following listing shows the result. READ.A.LINE is indeed short and fulfills all the needs I had. Backspace is the only editing character. When you type a printing character, it is displayed and the underline cursor advances. When you type a backspace, the cursor backs up and replaces what was previously at that position.

When you type a RETURN, the routine returns with the X-register pointing into the line buffer at the position where the RETURN is stored. If you type an ESCAPE, it nulls the line you just typed. The line is still in the buffer, but READ.A.LINE returns with the X-register equal to zero.

Inside READ.A.LINE I use the X-register to track position in the line buffer. I don't explicitly keep track of position on the screen, at least in absolute terms. Wherever the cursor is when you call READ.A.LINE is the place where the first cursor will be displayed. Since I do all moving around on the screen by simply printing out characters or backspacing, all screen positions are relative to the beginning position. You can use READ.A.LINE inside a small window if you like, and the input line will wrap nicely when it comes to the edge of the window.

Lines 1100-1120 handle the cursor by first printing an underline, and backspacing over it. Then when you type a character I will put it over the top of the cursor, and come back to line 1100 to display the next cursor.

Lines 1130-1150 are the standard way to read the keyboard when you don't want any firmware interference. (The Apple IIgs has more hardware bells and whistles here, with the Front Desk Processor and all, but this method will still work.)

When you type a backspace, I have to output a space to erase the cursor, then backspace twice and place a new cursor over the top of the character which was right before the old cursor. This is in lines 1350-1380. If the cursor is already at the beginning of the line, lines 1330-1340 will simply start over. Because of the way lines 1330 and 1340 are written, the maximum length line you can type and still use backspace is 128 characters.

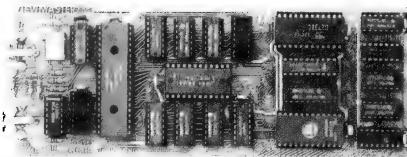
I wrote a quick and dirty test program, to illustrate how to use READ.A.LINE. Lines 1500-1560 print out a prompting message. Line 1580 calls READ.A.LINE. The cursor will start right after the prompting message. Line 1600 tests for an ESCAPE, signifying a null entry. If you type RETURN, lines 1610-1670 will display the message you just entered.

```

1000 *SAVE S.READLINE
1010 *
1020 KEYBOARD .EQ $C000
1030 STROBE .EQ $C010
1040 *
1050 CROUT .EQ $FD8E
1060 COUT .EQ $FDED
1070 *
1080 READ.A.LINE
0800- A2 00 1090 .0 LDX #0 Start at beginning of line
0802- A9 DF 1100 .1 LDA #"_ Use underline for a cursor
0804- 20 ED FD 1110 JSR COUT Print the Underline
0807- 20 49 08 1120 JSR BSOUT ...and back up over it
080A- AD 00 CO 1130 .2 LDA KEYBOARD
080D- 10 FB 1140 BPL .2 ...wait until key pressed
080F- 8D 10 CO 1150 STA STROBE
0812- 9D 4E 08 1160 STA WBUF,X
0815- C9 A0 1170 CMP #$A0
0817- 90 OA 1180 BCC .4 CONTROL CHAR
0819- E0 7F 1190 CPX #MAXLEN
081B- B0 ED 1200 BCS .2
081D- 20 ED FD 1210 JSR COUT SHOW IT ON SCREEN
0820- E8 1220 INX
0821- D0 DF 1230 BNE .1 ...ALWAYS
0823- C9 88 1240 .4 CMP #$88
0825- F0 OD 1250 BEQ .5 BACKSPACE
0827- C9 9B 1260 CMP #$9B
0829- F0 18 1270 BEQ .6 ESCAPE -- ABORT INPUT
082B- C9 8D 1280 CMP #$8D
082D- D0 DB 1290 BNE .2 ...NOT <RETURN>, IGNORE IT
082F- 20 46 08 1300 JSR SPCOUT ERASE CURSOR FROM SCREEN
0832- E8 1310 INX INCLUDE <RETURN> IN X-VALUE
0833- 60 1320 RTS RETURN TO CALLER
0834- CA 1330 .5 DEX
0835- 30 C9 1340 BMI .0 ...ALREADY AT BEGINNING OF LINE
0837- 20 46 08 1350 JSR SPCOUT ...SPACE
083A- 20 49 08 1360 JSR BSOUT ...BACKSPACE
083D- 20 49 08 1370 JSR BSOUT ...AND ANOTHER
0840- 4C 02 08 1380 JMP .1
0843- A2 00 1390 .6 LDX #0 SIGNAL ESCAPE TYPED
0845- 60 1400 RTS
1410 *
0846- A9 A0 1420 SPCOUT LDA #"_ SPACE
0848- 2C 1430 .HS 2C
0849- A9 88 1440 BSOUT LDA #$88 BACKSPACE
084B- 4C ED FD 1450 JMP COUT
1460 *
7F- 1470 MAXLEN .EQ 127
084E- 1480 WBUF .BS MAXLEN+1
1490 *
1500 TEST
08CE- A2 00 1510 LDX #0 PRINT OUT A PROMPT MESSAGE
08D0- BD F1 08 1520 .0 LDA PROMPT,X
08D3- 20 ED FD 1530 JSR COUT
08D6- E8 1540 INX
08D7- E0 0F 1550 CPX #PROMPT.LEN
08D9- 90 F5 1560 BCC .0
1570 *
08DB- 20 00 08 1580 JSR READ.A.LINE
1590 *
08DE- F0 10 1600 BEQ .2 ESCAPE ABORTED IT
08E0- 20 8E FD 1610 JSR CROUT <RETURN> OUTPUT
08E3- A2 00 1620 LDX #0 PRINT OUT THE LINE YOU TYPED IN
08E5- BD 4E 08 1630 .1 LDA WBUF,X
08E8- 20 ED FD 1640 JSR COUT
08EB- E8 1650 INX
08EC- C9 8D 1660 CMP #$8D UP TO A <RETURN>
08EE- D0 F5 1670 BNE .1
08F0- 60 1680 .2 RTS
1690 *
08F1- C5 EE F4
08F4- E5 F2 A0
08F7- E1 A0 EC
08FA- E9 EE E5
08FD- BA A0 A0 1700 PROMPT .AS -/Enter a line: /
0F- 1710 PROMPT.LEN .EQ #-PROMPT
1720 *

```

Now run AppleWorks™ on your II+ and keep full 80 column compatibility!



	80 COLUMN DISPLAY	80 COLUMN FORMAT	80 COLUMN DOS	80 COLUMN CP/M	7-9 PORT	80 COLUMN WORDSTAR	80 COLUMN FORMAT	80 COLUMN CP/M	80 COLUMN FORMAT
VIEWMASTER 80	✓	✓	✓	✓	✓	✓	✓	✓	✓
SLIPTERM	✓								
WIZARD 80				✓			✓		✓
VISION 80	✓	✓		✓					
OMNIVISION	✓					✓	✓		
VIEWMAX 80	✓	✓		✓			✓		
SMARTERM	✓	✓			✓	✓	✓		
VIDEOTERM	✓	✓	✓	✓	✓	✓	✓	✓	✓

One look at the chart will give you some of the reasons there's only one smart choice in 80 column cards for your Apple. But the real secret to Viewmaster 80's success is something even better: Total compatibility.

Each Viewmaster 80 includes our powerful Apple-Works expand software, allowing AppleWorks to run on the II+ with only 64K (or more) memory. (We recommend the RamFactor memory card, but any compatible Apple memory card will work.) The software provides our full range of AppleWorks enhancements, including expanded records, word processor, multi-disk saving, time and date display on screen with any PRO-DOS clock, and more!

The Viewmaster 80 works with all 80 column applications, including DOS 3.3, PRO-DOS, CP/M, Pascal, WordStar, Format II, Applewriter II, dBase II, VisiCalc, Multiplan, and *hundreds* of others — including AppleWorks.

Here are just a few of the powerful features the Viewmaster 80 delivers for a great price (\$149):

- 80 Characters by 24 lines • Fully compatible with all Apple languages and software • Highest compatibility with existing 80 column software • Very low power consumption • High speed (18 MHZ) scroll rate • Upper and lower case characters and true descenders, both inverse and normal; all on screen editing functions are supported • User-definable cursor shape • Compatible with Apple II, II+ and III • Five-year warranty

Call today to order or for more information, 9 a.m. to 11 p.m. seven days, or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5 1/4% sales tax. Add \$10.00 outside U.S.A.

CP/M is a registered trademark of Digital Research, Inc.

AE Applied Engineering
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060

Friendly Decimal Print Subroutine.....Jan Eugenides

Several years ago Bob S-C published a little subroutine to convert two-byte hexadecimal values to decimal and print them. I have written a similar subroutine, but mine allows you to print any character as a leading character when the number has less than five digits, or to have no leading characters at all. Bob's subroutine always printed leading zeroes.

To use my subroutine, you put the number in the A- and X-registers (lo-byte in A, hi-byte in X), and the fill character in the Y-register. If you do not want any fill character, call my subroutine with Y=0. To make calling it easier, Bob S-C constructed a macro named PDEC, defined in lines 1490-1580 below. To print a number with no fill character, simply use the macro like this: >PDEC VALUE; to print with a fill character, use the macro with the fill character as the second parameter like this: >PDEC VALUE,"#".

Now let's look at my subroutine. Lines 1110-1120 store the high byte of the number, and the fill character. I waited to store the low-byte of the number in line 1160, inside the conversion loop, to save a few bytes and cycles.

Lines 1150-1380 comprise a loop controlled by X, counting down from 3 to 0. The loop prints out the first four digits and/or fill characters. X indexes into a power-of-tens table. I first subtract 10000 as many times as possible, counting each subtraction in the Y-register. Since Y starts out as ASCII 0, it increments through successive ASCII digits. When the remainder is finally less than 10000, I print the first digit. Then the process is repeated for 1000, 100, and 10.

Leading-zeroes may be either suppressed, or printed with your choice of fill character. The work is done in lines 1280-1340. I keep a flag in bit 7 of FLAG. This bit starts out 0 (see line 1140). As soon as I get a non-zero digit, I change the flag bit to 1 (see lines 1280-1300). If the flag bit is 1, lines 1310-1320 will decide to print the digit regardless of its value. (We don't want to omit or replace ALL zeroes, only LEADING zeroes.) Lines 1330-1340 will decide whether to print a fill character or nothing at all.

After printing the ten's digit, the remainder in NUM is the units digit in binary. All I have to do is add an ASCII "0" to change this remainder to ASCII, which I do in lines 1390-1400. By handling the unit's digit in this special way I save extra logic which would have been required to make the value \$0000 print at least one zero and two bytes in my power-of-tens table, saving a total of six bytes. The cost is lines 1390 and 1400 versus a simple RTS, or four bytes. This method is also faster, because I don't have to cycle through the loop counting down the unit's digit.

For an exercise, you might try extending my subroutine to convert 3- or 4-byte values. It is really not too difficult. Of course, you would no longer be able to pass the value to be converted in the A- and X-registers. For 3-byte values, the



SPECIAL !!! EXPANDED RAM/ROM BOARD: \$39.00

Similar to our \$30 RAM/ROM dev board described below. Except this board has two sockets to hold your choice of 2-2K RAM, 2-2K ROM or even 2-4K ROM for a total of 8K. Mix RAM and ROM too. Although Apple limits access to only 2K at a time, soft switches provide convenient socket selection. Hard switches control defaults.

IMPROVED !!! IN A MAC (ver 2.0): \$75.00

Now includes faster graphics, UniDisk support and more! Bi-directional data transfers are a snap! This Apple II emulator runs DOS 3.3/PRODOS (including 6502 machine language routines) on a 512K MAC or MACPLUS. All Apple II features are supported such as HI/LO-RES graphics, 40/80 column text, language card and joystick. Also included: clock, RAM disk, keyboard buffer, on-screen HELP, access to the desk accessories and support for 4 logical disk drives. Includes 2 MAC diskettes (with emulation, communications and utility software, plus DOS 3.3 and PRODOS system masters, including Applesoft and Integer BASIC) and 1 Apple II diskette.

SCREEN.GEN: \$35.00

Develop HI-RES screens for the Apple II on a Macintosh. Use MACPAINT (or any other application) on the MAC to create your Apple II screen. Then use SCREEN.GEN to transfer directly from the MAC to an Apple II (with SuperSerial card) or IIC. Includes Apple II diskette with transfer software plus fully commented SOURCE code.

MIDI-MAGIC for Apple //c: \$49.00

Compatible with any MIDI equipped music keyboard, synthesizer, organ or piano. Package includes a MIDI-out cable (plugs directly into modem port - no modifications required!) and 6-song demo diskette. Large selection of digitized QRS player-piano music available for 19.00 per diskette (write for catalog). MIDI-MAGIC compatible with Apple II family using Passport MIDI card (or our own input/output card w/drum sync for only \$99.00).

FONT DOWNLOADER & EDITOR: \$39.00

Turn your printer into a custom typesetter. Downloaded characters remain active while printer is powered. Use with any Word Processor program capable of sending ESC and control codes to printer. Switch back and forth easily between standard and custom fonts. Special functions (like expanded, compressed etc.) supported. Includes HIRES screen editor to create custom fonts and special graphics symbols. For Apple II, II+, //e. Specify printer: Apple ImageWriter, Apple Dot Matrix, C.Itoh 8510A (Prowriter), Epson FX 80/85, or Okidata 92/192.

*** FONT LIBRARY DISKETTE #1: \$19.00** contains lots of user-contributed fonts for all printers supported by the Font Downloader & Editor. Specify printer with order.

DISASM 2.2e \$30.00 (\$50.00 with SOURCE Code)

Use this intelligent disassembler to investigate the inner workings of Apple II machine language programs. DISASM converts machine code into meaningful, symbolic source compatible with S-C, LISA, ToolKit and other assemblers. Handles data tables, displaced object code & even provides label substitution. Address-based triple cross reference generator included. DISASM is an invaluable machine language learning aid to both novice & expert alike. Don Lancaster says DISASM is "absolutely essential" in his ASSEMBLY COOKBOOK.

The 'PERFORMER' CARD: \$39.00 (\$59.00 with SOURCE Code)

Converts a 'dumb' parallel printer I/F card into a 'smart' one. Simple command menu. Features include perforation skip, auto page numbering with date & title, large HIRES graphics & text screen dumps. Specify printer: MX-80 with Graftrax-80, MX-100, MX-80/100 with Graftraxplus, NEC 8092A, C.Itoh 8510 (Prowriter), Okidata 82A/83A with Okigraph & Okidata 92/93.

'MIRROR' ROM: \$25.00 (\$45.00 with SOURCE Code)

Communications ROM plugs directly into Novation's Apple-Cat Modem card. Basic modes: Dumb Terminal, Remote Console & Programmable Modem. Features include: selectable pulse or tone dialing, true dialtone detection, audible ring detect, ring-back, printer buffer, 80 col card & shift key mod support.

RAM/ROM DEVELOPMENT BOARD: \$30.00

Plugs into any Apple slot. Holds one user-supplied 2Kx8 memory chip (6116 type RAM for program development or 2716 EPROM to keep your favorite routines on-line). Maps into \$Cn00-CnFF and \$C800-CFFF.

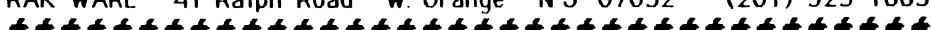
C-PRINT For The APPLE //c: \$69.00

Connect standard parallel printers to an Apple //c serial port. Separate P/S included. Just plug in and print!

Unless otherwise specified, all Apple II diskettes are standard (not copy protected!) 3.3 DOS.

Avoid a \$3.00 handling charge by enclosing full payment with order. VISA/MC and COD phone orders OK.

RAK-WARE 41 Ralph Road W. Orange NJ 07052 (201) 325-1885



power-of-ten tables would have to be extended in both dimensions:

```
TENTBL .DA #10,#100,#1000,#10000
        .DA #100000,#1000000,#10000000
TENTBM .DA /10,/100,/1000,/10000
        .DA /100000,/1000000,/10000000
TENTBH .DA #0,#0,#0,#0
        .DA /100000/256,/1000000/256,/10000000/256
```

You would also have to extend the comparison and subtraction code inside the loop. Why not try it?

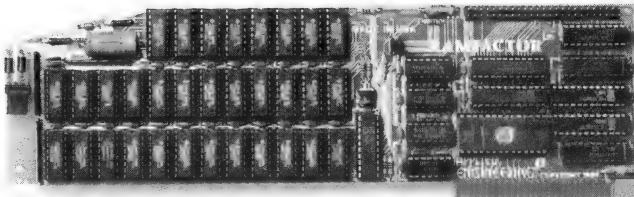
```
1000 *SAVE S.PRINT.DECIMAL
1010 *
1020 * By Jan Eugenides and Bob S-C
1030 *
1040 * Call with A, X, and Y as follows:
1050 * (A) = low-byte of number to be printed
1060 * (X) = high byte of number
1070 * (Y) = fill character (or 00 if no fill)
1080 *
FDED- 1090 COUT .EQ $FDED
1100 *
0800- 8E 52 08 1110 PRDEC STX NUM+1 Store high byte of number
0803- 8C 50 08 1120 STY FILL Store fill character
0806- A2 03 1130 LDX #3 FOR X = 3 TO 0
0808- BE 4F 08 1140 STX FLAG Clear bit 7 in leading-zero flag
080B- A0 B0 1150 .1 LDY "#0" Start with digit = ASCII 0
080D- 8D 51 08 1160 .2 STA NUM Compare number to power of ten
0810- DD 47 08 1170 CMP TENTBL,X  $10^{(X+1)}$  ... lo-byte
0813- AD 52 08 1180 LDA NUM+1
0816- FD 4B 08 1190 SBC TENTBH,X  $10^{(X+1)}$  ... hi-byte
0819- 90 0C 1200 BCC .3 Remainder is smaller than  $10^{(X+1)}$ 
081B- 8D 52 08 1210 STA NUM+1 Store remainder hi-byte
0821- FD 47 08 1220 LDA NUM Get remainder lo-byte
0824- C8 1240 INY Increment ASCII digit
0825- D0 E6 1250 BNE .2 ... always
1260 *----Print a digit-----
0827- 98 1270 .3 TYA digit in ASCII
0828- C9 B0 1280 CMP "#0" Is it a zero?
082A- F0 03 1290 BEQ .4 ...yes, might be leading zero
082C- 8D 4F 08 1300 STA FLAG ...no, so clear leading-zero flag
082F- 2C 4F 08 1310 .4 BIT FLAG If this is leading-zero, will be +
0832- 30 05 1320 BMI .5 ...not a leading zero
0834- AD 50 08 1330 LDA FILL ...leading zero, so use fill-char
0837- F0 03 1340 BEQ .6 ...Oops, no fill-char
0839- 20 ED FD 1350 .5 JSR COUT Print the digit or fill-char
083C- AD 51 08 1360 .6 LDA NUM Get lo-byte of remainder
083F- CA 1370 DEX Next X
0840- 10 C9 1380 BPL .1 Go get next digit
0842- 09 B0 1390 ORA "#0" Change remainder to ASCII
0844- 4C ED FD 1400 JMP COUT Print Unit's digit & RTS
1410 *
0847- 0A 64 E8
084A- 10 1420 TENTBL .DA #10,#100,#1000,#10000
084B- 00 00 03
084E- 27 1430 TENTBH .DA /10,/100,/1000,/10000
1440 *
084F- 1450 FLAG .BS 1
0850- 1460 FILL .BS 1
0851- 1470 NUM .BS 2
1480 *
1490 *----ma pdec
1500 lda #1
1510 ldx #1+1
1520 .do #1>1
1530 ldy "#12"
1540 .else
1550 ldy #0
1560 .fin
1570 jsr prdec
1580 .em
```

RamFactor™

Boot any II+, IIe, or IIgs and be up and running your favorite software in less than 1 second.

Now anyone with an Apple II+, Franklin, Laser 128, Apple IIe or IIgs preferring to use slots 1 through 7 can now enjoy the speed and performance that they've been waiting for.

With RamFactor, you'll be able to instantly add another 256K, 512K, or a full 1 MEG on the main board and an expansion connector can address an additional 16 MEG of memory for possible future expansion. Since virtually all software is automatically compatible with RamFactor, you'll immediately be able to load programs into RamFactor for instantaneous access to information. You'll also be able to store more data for larger word processing documents, bigger data bases, and expanded spreadsheets. And with the battery back-up option, you can switch on your Apple and your favorite software will be up and running in less than 1 second!



Very Compatible

All the leading software is already compatible with RamFactor. Programs like AppleWorks, Pinpoint, BPI, Managing Your Money, Dollars and Sense, SuperCalc 3A, PFS, MouseWrite, MouseDesk, MouseCalc, Sensible Speller, Applewriter IIe, Business Works, ReportWorks, Catalyst 3.0 and more. And RamFactor is fully ProDOS, DOS 3.3, Pascal 1.3 and CP/M compatible. In fact, no other slot 1-7 memory card is more compatible with commercial software.

AppleWorks Power

There are other slot 1-7 cards that give AppleWorks a larger desktop, but that's the end of their story. But RamFactor is the only slot 1-7 card that increases AppleWorks internal memory limits, increasing the maximum number of records in the database and lines permitted in the word processor, and RamFactor is the only standard slot card that will automatically load all of AppleWorks into RAM dramatically increasing speed and eliminating the time required to access the program disk. It will even display the time and date on the AppleWorks screen with any ProDOS clock. RamFactor will automatically segment large files so they can be saved on 5 1/4", 3 1/2", and hard disks. All this performance is available on the Apple II+,

IIe, Franklin, or Laser 128 when used with an 80 column display. RamFactor, no other standard slot card comes close to enhancing AppleWorks so much.

The "Instant On" Apple.



With the RamCharger battery backup option, RamFactor will retain programs and data during power failures or even

when you turn your computer off. And because RamFactor plugs into slots 1-7, it's the only memory card that can cold boot an Apple without using slow floppy or hard disks. An available "Y" cable allows one RamCharger to power two RamFactors.

Powerful Program Switcher

With RamFactor, you can organize memory into multiple work areas and switch between them. Each work area can contain different programs and even different operating systems. Now you can switch from one program to another or even switch from AppleWorks to DOS 3.3 to CP/M to Pascal to ProDOS in under a second. And with our battery back-up option, you can have permanent storage for up to 20 years.

Features:

- 256K to 1 MEG on main board. Expansion connector for possible further expansion to 16 MEG.
- Fully Apple II Memory Expansion compatible
- Compatible with Apple IIe, II+, IIgs, Franklin and Laser 128
- Battery back-up option allows you to turn on your Apple and run your favorite programs in less than 1 second!
- Automatically recognized by ProDOS, DOS 3.3, Pascal, and standard CP/M
- Built-in RamDrive™ software (a true RAM disk not disk caching)
- Systems are directly bootable from RamFactor if desired
- Built in self diagnostic software
- Automatic expansion with AppleWorks 1.3 or 2.0
- Allows Apple II+ and IIe to run your AppleWorks without buying any additional software
- Accelerates AppleWorks
- Displays time and date on the AppleWorks screen with any ProDOS clock
- Fits any I/O slot except slot 3
- Fully socketed and user upgradeable
- Much, much more

RamFactor with 256K	\$239
RamFactor with 512K	\$279
RamFactor with 1 MEG	\$359
RamFactor with 2-16 MEG	CALL
RamCharger backup option	\$179
"Y" cable	\$24

(Allows one RamCharger to power two RamFactors.)

Order RamFactor today with 15 day money back guarantee and our "no hassle" five year warranty. See your dealer or call (214) 241-6060, 9 a.m. to 11 p.m., 7 days or send check or money order to Applied Engineering. MasterCard, Visa and C.O.D. welcome. Texas residents add 6 1/4% sales tax. Add \$10.00 if outside USA.

AE Applied Engineering™
The Apple enhancement experts

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

FD8E-	1590	*	-----
	1600	CROUT	.EQ \$FD8E
	1610		
0853-	1620	T	>PDEC VAL1, #
0853-	0000>	lda VAL1	
0856-	AD 9A 08	0000>	ldx VAL1+1
	0000>		.do 2>1
0859-	AE 9B 08	0000>	ldy #*#
	0000>		.else
	0000>		.fin
085B-	20 00 08	0000>	jsr prdec
085E-	20 8E FD	1630	JSR CROUT
0861-		1640	>PDEC VAL2, ^
0861-	AD 9C 08	0000>	lda VAL2
0864-	AE 9D 08	0000>	ldx VAL2+1
	0000>		.do 2>1
0867-	AO DE	0000>	ldy #*#
	0000>		.else
	0000>		.fin
0869-	20 00 08	0000>	jsr prdec
086C-	20 8E FD	1650	JSR CROUT
086F-		1660	>PDEC VAL3
086F-	AD 9E 08	0000>	lda VAL3
0872-	AE 9F 08	0000>	ldx VAL3+1
	0000>		.do 1>1
	0000>		.else
0875-	AO 00	0000>	ldy #0
	0000>		.fin
0877-	20 00 08	0000>	jsr prdec
087A-	20 8E FD	1670	JSR CROUT
087D-		1680	>PDEC VAL4
087D-	AD A0 08	0000>	lda VAL4
0880-	AE A1 08	0000>	ldx VAL4+1
	0000>		.do 1>1
	0000>		.else
0883-	AO 00	0000>	ldy #0
	0000>		.fin
0885-	20 00 08	0000>	jsr prdec
0888-	20 8E FD	1690	JSR CROUT
088B-		1700	>PDEC VAL5, " "
088B-	AD A2 08	0000>	lda VAL5
088E-	AE A3 08	0000>	ldx VAL5+1
	0000>		.do 2>1
0891-	AO A0	0000>	ldy #*#
	0000>		.else
	0000>		.fin
0893-	20 00 08	0000>	jsr prdec
0896-	20 8E FD	1710	JSR CROUT
0899-	60	1720	RTS
	1730		
089A-	D2 04	1740	VAL1 .DA 1234
089C-	4F 00	1750	VAL2 .DA 79
089E-	FF FF	1760	VAL3 .DA 65535
08A0-	4B 1B	1770	VAL4 .DA 6987
08A2-	17 00	1780	VAL5 .DA 23

Note from American Foundation for the Blind

The American Foundation for the Blind is managing a database of individuals with hands-on experience with computers, low vision aids, talking products, and other devices for the blind and visually-impaired. About 500 users and evaluators are currently listed in their database, and are available to help in training, purchase decisions, and evaluation of new products. If you have related experience are interested in helping, or could use such help yourself, you can contact their National Technology Center at 1-800-232-5463.

To boldly go at speeds no man has gone before.

With the TransWarp™ accelerator, your IIe™ or II+™ is 40% faster than the IIGs!

Computing at warp speed!

It's an experience you shouldn't miss. And with TransWarp, you won't have to. Because TransWarp will run your IIe or II+ software 3.6 times faster — that's 40% faster than Apple's IIGs.

No more yawning while your Apple™ slowly rearranges text or calculates spreadsheets. With 256K of ultra fast RAM, TransWarp speeds up *all* Apple software — including AppleWorks, and all educational software, graphics and games. And it's compatible with all standard peripheral cards (such as RAM Works and Apple memory cards), hard disks, 5 1/4" 1mb disks, 80-column cards, modems, clock cards, mouses and more! You name it, TransWarp accelerates it. There's even a 16 bit upgrade chip available.



"TransWarp is great! I have replaced all my other accelerators with it!"

Steve Wozniak, the creator of Apple computers

An important difference.

With TransWarp, software runs up to three times faster than with other cards, since the others can't accelerate programs in auxiliary memory. That's why TransWarp is so much faster than the rest. Nearly all of today's more powerful programs run partially or completely in auxiliary memory programs like AppleWorks, Pinpoint, Managing Your Money, SuperCalc 3+, BPL and Pascal, to name a few. Why settle for a card that accelerates only part of the memory?

There's one more important difference. Since TransWarp doesn't use memory caching, you get *consistent* high speed performance.

A cinch to use.

Simply plug TransWarp into any slot in your Apple II, II+ or IIe — including slot 3 in the IIe. Instantly you'll be computing

at speeds you only dreamed about before. And should you ever wish to run at normal speed, simply press the ESC key while turning your Apple on.

Since TransWarp is completely transparent, you won't need preboot disks or special software. It's ready to go right out of the package!

Speed = Productivity

Imagine the productivity gains you'll achieve when your programs are running over three times faster. TransWarp is so powerful, your Apple will make IBM PCs™ and even ATs™ look like slowpokes.

- 3.6 MHZ 65C02
- 256K of ultra fast on-board RAM
- Accelerates main *and* auxiliary memory
- Low power consumption for cool operation
- totally transparent operation with all software
- Plugs into any slot, including slot 3 on the Apple IIe
- Accelerated 16 bit option available
- 5-year warranty

Satisfaction guaranteed!

Give your Apple the TransWarp advantage. With our risk-free 15 day money back guarantee, you have nothing to lose but wasted time. Call today!

TransWarp accelerator \$279
16 bit upgrade (may add later) \$89

For fast response:

Call Applied Engineering, 9 a.m. to 11 p.m., 7 days at (214) 211-0000. MasterCard, VISA and C.O.D. welcome. Texas residents add 5 1/2% sales tax. Add \$10.00 if outside U.S.A. Or mail check or money order to Applied Engineering, P.O. Box 798, Carrollton, TX 75006.

AE Applied Engineering
The Apple enhancement experts.

P.O. Box 798, Carrollton, TX 75006 (214) 241-6060

Some Handy MOVE Macros.....Bob Sander-Cederlof

Some time ago Bruce Love, a mathematics teacher at Hillcrest High School in New Zealand, sent me a macro he has found quite handy. It is shown in lines 1010-1090 in the listing below. It can be used to generate in-line loops to copy a group of bytes from one place to another. The macro works with any group up to 256 bytes long. It is really quite flexible, as the several examples in lines 1460-1550 reveal. Note that the source and destination may be either absolute or indirect in form. Furthermore, the length may either be stored in a variable or in the immediate form.

I added the SET macro (lines 1170-1230) to set up a pointer, to simplify using the MOVE macro with pointers in page zero. An example of its use is in lines 1490 and 1500.

There is another macro that I have found useful for moving larger chunks of RAM around. I use it especially in ProDOS SYSTEM files for relocating program sections. See lines 1100-1160 for the definition of the COPY macro, which moves up to 256 pages from source to destination. The parameters for COPY are reminiscent of the monitor M-command. For example, to copy \$4B00 through \$5FFF to a destination beginning at \$6A00 the monitor command would be "6A00<4B00.5FFF". The equivalent COPY macro would be "COPY \$6A00,\$4B00,\$5FFF".

With both of these macros, beware of overlapping the range of the source and destination just as you would with the monitor M-command. Unless, of course, you know what you are doing and are doing it on purpose. For example, you can copy the first character of a block to all positions in the block like this:

```
>MOVE BUFFER,BUFFER+1,LEN
```

As is usually the case with macros which generate code, the code will probably not be as efficient as you could write given more knowledge of the overall situation. However, macros like these can shorten the time to develop programs, and thus they are worthy in spite of their possible inefficiencies.

```
1000 *SAVE MOVE.MACROS
1010 *-----
1020     .MA MOVE source,destination,length
1030     LDY #0
1040 :1    LDA ]1,Y      GET SOURCE BYTE
1050     STA ]2,Y      STORE DEST. BYTE
1060     INY             NEXT BYTE
1070     CPY ]3
1080     BCC :1          ...IF MORE
1090     .EM
1100 *
1110     .MA COPY destination,src.start,src.end
1120     LDA /]1           DESTINATION PAGE
1130     LDY /]2           SOURCE BEGINNING PAGE
1140     LDX /]3-]2+256    # OF PAGES
1150     JSR COPY.PAGES
1160     .EM
1170 *
1180     .MA SET pointer,value
1190     LDA #]2
1200     STA ]1
1210     LDA /]2
1220     STA ]1+1
1230     .EM
```

```

00-          1240 *-----+
02-          1250 SPTR .EQ $00,01
04-          1260 DPTR .EQ $02,03
0200-        1270 LENGTH .EQ $04
0200-        1280 BUFFER .EQ $200
0200-        1290 *
1300 COPY.PAGES
0800-  85 03  1310 STA DPTR+1
0802-  84 01  1320 STY SPTR+1
0804- A0 00  1330 LDY #0
0806-  84 02  1340 STY DPTR
0808-  84 00  1350 STY SPTR
080A- B1 00  1360 .1 LDA (SPTR),Y
080C-  91 02  1370 STA (DPTR),Y
080E- C8    1380 INY      NEXT BYTE
080F- D0 F9  1390 BNE .1   ...KEEP GOING
0811- E6 01  1400 INC SPTR+1 ...END OF PAGE
0813- E6 03  1410 INC DPTR+1
0815- CA    1420 DEX      NEXT PAGE
0816- D0 F2  1430 BNE .1   ...KEEP GOING
0818-  60    1440 RTS     ...FINISHED
0818-        1450 *-----+
1460 SAMPLE.MOVES
0819-        1470 >MOVE $500,$600,#40
0819- A0 00  0000> 0000> LDY #0
081B- B9 00  0000> :1  LDA $500,Y   GET SOURCE BYTE
081E- 99 00  0000> 0000> STA $600,Y   STORE DEST. BYTE
0821- C8    0000> INY      NEXT BYTE
0822- C0 28  0000> CPY #40
0824- 90 F5  0000> BCC :1   ...IF MORE
0824-        1480 *-----+
0826-        1490 >SET SPTR,BUFFER+10
0826- A9 0A  0000> LDA #BUFFER+10
0828- 85 00  0000> STA SPTR
082A- A9 02  0000> LDA /BUFFER+10
082C- 85 01  0000> STA SPTR+1
082E-        1500 >SET DPTR,BUFFER+40
082E- A9 28  0000> LDA #BUFFER+40
0830- 85 02  0000> STA DPTR
0832- A9 02  0000> LDA /BUFFER+40
0834- 85 03  0000> STA DPTR+1
0836- A9 09  1510 LDA #9
0838- 85 04  1520 STA LENGTH
083A- A0 00  0000> >MOVE (SPTR),(DPTR),LENGTH
083C- B1 00  0000> :1  LDA (SPTR),Y   GET SOURCE BYTE
083E- 91 02  0000> STA (DPTR),Y   STORE DEST. BYTE
0840- C8    0000> INY      NEXT BYTE
0841- C4 04  0000> CPY LENGTH
0843- 90 F7  0000> BCC :1   ...IF MORE
0843-        1540 *-----+
0845-        1550 >MOVE BUFFER,(DPTR),#10
0845- A0 00  0000> LDY #0
0847- B9 00  0000> :1  LDA BUFFER,Y   GET SOURCE BYTE
084A- 91 02  0000> STA (DPTR),Y   STORE DEST. BYTE
084C- C8    0000> INY      NEXT BYTE
084D- C0 0A  0000> CPY #10
084F- 90 F6  0000> BCC :1   ...IF MORE
084F-        1560 *-----+
1570 SAMPLE.COPIES
0851-        1580 >COPY $6A00,$4B00,$5FFF
0851- A9 6A  0000> LDA /$6A00   DESTINATION PAGE
0853- A0 4B  0000> LDY /$4B00   SOURCE BEGINNING PAGE
0855- A2 15  0000> LDX /$5FFF-$4B00+256 # OF PAGES
0857- 20 00  08 0000> JSR COPY.PAGES
085A-        1590 >COPY $5000,$2200,$4AFF
085A- A9 50  0000> LDA /$5000   DESTINATION PAGE
085C- A0 22  0000> LDY /$2200   SOURCE BEGINNING PAGE
085E- A2 29  0000> LDX /$4AFF-$2200+256 # OF PAGES
0860- 20 00  08 0000> JSR COPY.PAGES
0863-        1600 >COPY $1000,BUFFER,BUFFER+255
0863- A9 10  0000> LDA /$1000   DESTINATION PAGE
0865- A0 02  0000> LDY /BUFFER   SOURCE BEGINNING PAGE
0867- A2 01  0000> LDX /BUFFER+255-BUFFER+256 # OF PAGES
0869- 20 00  08 0000> JSR COPY.PAGES
086C- 60    1610 RTS
086C-        1620 *-----+

```

New Supplement to "Beneath Apple ProDOS" Available

Quality Software has published a new supplement to for "Beneath Apple ProDOS," which includes information on ProDOS versions 1.2 and 1.3. It is 30 pages longer than the previous edition, which covered version 1.1.1 of ProDOS.

You might be wondering, "What is a supplement, anyway?" The book "Beneath Apple ProDOS" ("BAP") contains much reference material needed to really take advantage of ProDOS capabilities. While other books now cover much of the same ground, "BAP" was the first one to put it all into print. The supplement, however, is unique: it contains a complete description of the internal details of both the ProDOS MLI kernel and BASIC.SYSTEM. If you are at all involved with the inner works of ProDOS, or are having trouble finding out the REAL scoop on some issues, you NEED the supplement. I have used my copies extensively, and depended heavily on it when writing the ProDOS version of S-C Macro Assembler.

The original supplement, for versions 1.0.1 and 1.0.2 cost \$10. The second and third editions are \$12.50 each. Incredibly low-priced! You must order these directly from Quality Software, at 21610 Lassen Street #7, Chatsworth, CA 91311. As I understand it, the supplement is only sold to owners of "BAP", and you have to use the coupon found on page 8-9 of "BAP" to do the ordering. ("BAP" itself is \$19.95 retail, but we sell it for \$18 here.)

RAMWORKS™

ACCEPT NO SUBSTITUTES. BECAUSE THERE AREN'T ANY.

There's only one card like RamWorks. We've got the best hardware design. We supply the best software and we've got the best support from software companies.

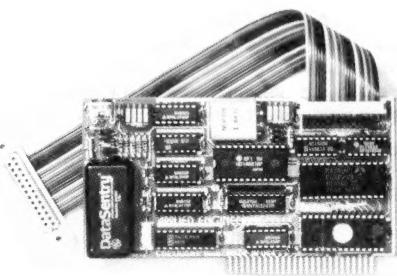
If someone tempts you with an imitation, please get both sides of the story. You'll discover why RamWorks offers the best enhancements to AppleWorks and other programs, and at the lowest price.

GUARANTEED!

214-241-6060
9 AM - 11 PM


"We Set the Standard"
APPLIED ENGINEERING

Serial Pro® The intelligent multifunction card from Applied Engineering.



Serial Pro is a powerful multifunction card for the Apple IIe, II+, IIgs and compatible computers. Serial Pro combines a powerful serial port for interfacing to a printer, modem or other serial device, with a full function clock/calendar. Serial Pro's serial port is compatible with virtually all letter quality and dot matrix printers, including Apple's ImageWriter and ImageWriter II, and can communicate with the fastest external modems at baud rates from 50 to 19,200.

Set up is easy, because unlike other cards, there are no switches to program serial parameters or disks to use. Serial Pro has a built-in setup screen that allows all parameters to be set. And on screen help that makes it super easy. An on-board battery stores your selections for over 20 years, but you can change them at any time. Serial Pro is the perfect upgrade for Super Serial card owners looking for more performance or a built-in clock.

Serial Port Features • Fully Super Serial Card compatible but more versatile and easier to use • Compatible with printers • Compatible with modems • Prints graphics to printer (without software) in normal mode, 90° rotation, double size, side by side, and black/white inversion • Variable word length, number of stop bits and parity selection • Crystal controlled circuits are used to provide ultra accurate baud rates

Clock Calendar Features • ProDOS, DOS and AppleWorks compatible. Displays time and date on the AppleWorks screen • Automatically time and date stamps AppleWorks files • 20 year auto recharging nickel-cadmium battery • Realtime interrupts • Built-in on screen time and date settings (no software needed) • Does not use phantom slots

Serial Pro is perfect for Apple IIe, II+, IIgs or compatible owners in need of a serial port or clock/calendar or both. But unlike other multifunction cards on the market, Serial Pro does not use phantom slots, so all the slots in your computer are useable.

Serial Pro comes complete with manual, cable and instructions for connecting to all the leading printers and modems.

Serial Pro **\$159**

Order Today! See your dealer or call Applied Engineering, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5% sales tax. Add \$10.00 outside U.S.A.

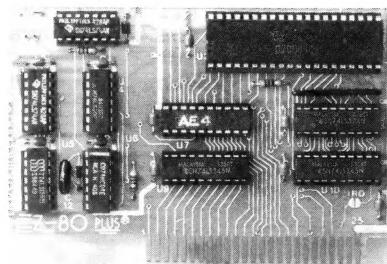
AE **APPLIED ENGINEERING™**

The Apple enhancement experts

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

With the Z-80 Plus®, run over 5000 new CP/M® programs.



*Now, get two computers
in one, and all the
advantages of both.*

Enter the CP/M world with the new Z-80 Plus card from Applied Engineering and introduce your Apple IIe, IIgs or II+ to thousands of CP/M programs. Only the Z-80 Plus comes packed with the new 5.1 software, the most advanced system ever for running CP/M programs.

The new 5.1 boasts advanced features like built-in disk emulation for popular memory expansion boards, boosting both system speed and storage capacity. And menu-driven utilities that let you get to work faster. The Z-80 Plus also lets you run older CP/M programs - all the way down to Version 1.6 (2.2 is the most popular).

The Z-80 Plus is the only card on the market capable of accessing more than 64K. If you have an extended 80-column card, 128K is usable. And if you have RamWorks, RamFactor, oSRam, oSRam Plus, or an Apple memory card in your IIe, IIgs or II+, up to 1088K is available.

Each Z-80 Plus comes with our CP/M Ram Drive software, enabling most memory cards to work as a high speed Ram disk so CP/M software can run up to *twenty times faster*. So packages like WordStar, dBASE II, Turbo Pascal and most others run at blinding speed.

Simply plug the Z-80 Plus into any slot in your Apple. You'll get the benefits of two computers in one - all at an unbelievably low price (only \$149).

• Fully compatible with ALL CP/M software • Fully compatible with 5 1/4" and 3 1/2" disk drives as well as most hard disks including the Sider • Fully compatible with Microsoft disks (no pre-boot required) • Specifically designed for high speed operation • Runs WordStar, dBASE II, Turbo Pascal, and ALL other CP/M software with *no pre-boot* • Semi custom IC and low parts count allows Z-80 Plus to fly through CP/M programs with extremely low power consumption (we use the Z-80B) • Does EVERYTHING other Z-80 boards do, *plus* Z-80 interrupts • Five year warranty

To get your Z-80 Plus or for more information see your dealer, or call 214-241-6060 9 a.m. to 11 p.m. seven days, or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5% sales tax. Add \$10.00 outside U.S.A.

AE **Applied Engineering™**

P.O. Box 798, Carrollton, TX 75006

(214) 241-6060

CP/M is a registered trademark of Digital Research

A Note about Chessmaster 2000

Gina Montagano, of the Software Toolworks, recently sent me "Chessmaster 2000". If you have not heard of it, Chessmaster is a really good chess-playing program which will run on the Apple II family. It is also available for Mac, IBM, and others. I bought Microchess back in 1977, and Sargon in 1978; they did well, considering they ran in such small machines. Microchess would fit in a 4K Apple! We have come a long way since then. Chessmaster has many nice features, is the current reigning "champion", and is well worth the \$49.95 price. (My 11-year-old stepson Matthew figured out a neat way to beat it. He let it tramp all over him, all the way to checkmate. Then he stepped back one move, swapped sides, and won!) Some features: display board in color or monochrome, 2-D or 3-D; 20 levels of competence, plus one that purposely makes occasional obviously bad moves; optional hints for your best move; huge opening book; library of famous games; and on and on. Negative: copy protected, although you may order a backup copy for \$5. Chessmaster 2000 is available at most software stores.

EnterSoft

Basic-like macros for the S-C Macro Assembler.

Available for the Apple II, II+, //e, and //c.

Uses whatever DOS your S-C Macro Assembler uses.

Soon to be released for PRODOS.

Source code included with every disk!!!

Cost? A paltry \$30.00!!!

To Order - Send Check or Money Order to:



Mark Manning
c/o Baggy Games
P. O. Box 591894
Houston, Texas 77259-1894

Apple is a registered Trademark of Apple Computer Inc.

The S-C Macro Assembler is produced by The S-C Software Corporation.

EnterSoft is produced by Mark Manning/Simulacron I/Baggy Games

Apple Assembly Line (ISSN 0889-4302) is published monthly by S-C SOFTWARE CORPORATION, P.O. Box 280300, Dallas, Texas 75228. Phone (214) 324-2050. Subscription rate is \$18 per year in the USA, sent Bulk Mail; add \$3 for First Class postage in USA, Canada, and Mexico; add \$14 postage for other countries. Back issues are \$1.80 each (other countries inquire for postage). A subscription to the newsletter and the Monthly Disk containing all source code is \$64 per year in the US, Canada and Mexico, and \$87 to other countries.

All material herein is copyrighted by S-C SOFTWARE CORPORATION, all rights reserved. (Apple is a registered trademark of Apple Computer, Inc.)